

**IMPLEMENTASI *BACKGROUND SUBTRACTION* UNTUK  
KLASIFIKASI KERIPIK KENTANG BERBASIS RASPBERRY PI  
MENGUNAKAN METODE *NAIVE BAYES***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun oleh:  
Yongki Pratama  
NIM: 145150301111021



PROGRAM STUDI TEKNIK KOMPUTER  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## LEMBAR PENGESAHAN

IMPLEMENTASI *BACKGROUND SUBTRACTION* UNTUK KLASIFIKASI KERIPIK  
KENTANG BERBASIS RASPBERRY PI MENGGUNAKAN METODE *NAÏVE BAYES*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun Oleh :  
Yongki Pratama  
NIM: 145150301111021


Skripsi ini telah diuji dan dinyatakan lulus pada  
3 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

  
Dr. Eng. Fitri Utaminingrum, S.T., M.T.  
NIP. 19820710 200812 2 001

  
Wijaya Kurniawan, S.T., M.T.  
NIP. 19820125 201504 1 002

Mengetahui  
Ketua Jurusan Teknik Informatika



  
Tri Astoto Kurniawan, S.T., M.T., Ph.D  
NIP. 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Penulis menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 7 Juli 2018



Yongki Pratama

NIM: 145150301111021

## KATA PENGANTAR

Puja dan puji syukur atas kehadiran Allah SWT karena berkat rahmat serta hidayah-Nya, penulis dapat menyelesaikan penelitian untuk memperoleh gelar Sarjana Teknik yang berjudul “Implementasi *Background Subtraction* Untuk Klasifikasi Keripik Kentang Berbasis Raspberry Pi Menggunakan Metode Naïve Bayes”.

Dalam penyusunan dan penulisan skripsi ini tidak lepas dari bantuan-bantuan yang diberikan oleh berbagai pihak, maka penulis mengucapkan banyak-banyak terima kasih kepada :

1. Keempat orangtua saya atas segala nasehat, semangat dan dorongan baik berupa do’a maupun materi selama penulis melakukan penelitian.
2. Ibu Ambarwati, S.Sn dan Bapak Ary Yuspita atas segala nasehat, motivasi, semangat dan do’a selama penulis melakukan penelitian.
3. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Deskan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
5. Bapak Sabriansyah Rizqika Akbar, S.t., M.Eng. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
6. Ibu Dr. Eng.Fitri Utaminingrum, S.T, M.T selaku dosen pembimbing I yang telah memberikan pengarahan dan bimbingan, sehingga dapat menyelesaikan skripsi ini dengan baik.
7. Bapak Wijaya Kurniawan, S.T, M.T selaku dosen pembimbing II yang telah memberikan pengarahan dan bimbingan kepada peneliti, sehingga dapat menyelesaikan skripsi ini dengan baik.
8. Teman-teman “D-GENGZ” dan kontrakan ceria yang selalu membantu dan memberikan semangat serta motivasi selama proses pengerjaan skripsi ini.
9. Putri Jania S. yang selalu memberi dorongan, membantu memberi semangat serta motivasi selama pengerjaan skripsi.
10. Rekan-rekan pengurus Unitantri, Sabrangan wetan, Sanggar Budaya Nusantara dan seluruh teman-teman Teknik Komputer 2014 yang selalu memotivasi penulis selama pengerjaan skripsi ini.

Penulis menyadari dalam penyusunan skripsi ini terdapat banyak kekurangan, sehingga saran dan kritik yang membangun sangat dibutuhkan untuk ke depannya agar lebih baik lagi. Semoga penelitian ini dapat memberikan manfaat bagi semua pihak yang membacanya.

Malang, 18 Juli 2018

Yongki Pratama

Yongkipratama114@gmail.com

## ABSTRAK

Perkembangan ilmu pengetahuan dan teknologi pada masa sekarang banyak digunakan didunia industri untuk pengolahan hasil pertanian dan perkebunan. Kentang adalah jenis sayuran yang bisa diolah menjadi beragam makanan, salah satunya ialah keripik. Banyak perusahaan besar di Indonesia yang memproduksi keripik dari kentang. Salah satunya yaitu Pabrik Istana yang berada di Kota Wisata Batu. Kentang yang diolah mempunyai beragam ukuran mulai dari ukuran yang paling besar yaitu (Super), lalu ukuran (AB) yang berukuran sedang, kemudian ukuran (A) yang paling kecil. Proses penyortiran pada pabrik tersebut dilakukan oleh manusia secara manual sehingga menghasilkan keluaran yang relatif kurang seragam, Oleh karena itu dibutuhkan penelitian mengenai suatu alat yang dapat menyortir atau memilah keripik kentang secara otomatis. Dalam penelitian ini, sistem yang dibuat berupa konveyor, yang dipasang sebuah webcam sebagai sensor untuk mengambil gambar dari keripik kentang yang kemudian diolah dalam Raspberry Pi menggunakan metode pengolahan citra *Background Subtraction*. *Background Subtraction* adalah metode yang banyak digunakan untuk mendeteksi objek bergerak yang mana pada sistem ini keripik yang diambil nilainya adalah objek bergerak. Keripik kentang akan diklasifikasi berdasarkan ukuran yang dibaca oleh sistem dengan menggunakan parameter nilai  $w$  dan  $h$  atau lebar dan tinggi dari boundingbox keripik kentang yang dikonversi ke ukuran sebenarnya dengan satuan millimeter. Nilai tersebut digunakan sebagai acuan untuk diklasifikasi menggunakan metode *Naïve Bayes*. *Naïve Bayes* digunakan untuk metode klasifikasi karena merupakan metode yang mempunyai peforma tinggi dan mempunyai akurasi yang sangat baik untuk klasifikasi. Setelah mendapatkan hasil klasifikasi keripik kentang akan didorong atau dipilah oleh pendorong yang dijalankan servo. Dari hasil pengujian yang dilakukan *Background Subtraction* dapat membaca citra kentang dengan baik. Pembacaan ukuran keripik kentang dari sistem mendapatkan *error* yang kecil yaitu sebesar 3,73%. Kemudian Akurasi yang diperoleh menggunakan metode *Naïve Bayes* dalam klasifikasi keripik dengan 90 data latih dan 30 data uji adalah senilai 93,33 mempunyai kecepatan waktu pemrosesan rata-rata sebesar 1,7 ms dari 30 kali pengujian tersebut. Kemudian dilakukan pengujian hardware servo yang sudah berjalan sesuai sistem.

**Kata kunci:** keripik kentang, *Background Subtraction*, klasifikasi, *Naïve Bayes*



## ABSTRACT

Nowadays, the development of science and technology is widely used in the industrial world for the processing of plantation and agricultural products. One of the processings in processing of plantation and agricultural products is product election based on quality, such as ripeness level based on color, weight or size. Potatoes are a kind of vegetables that can be processed into various foods, one of which is the chips. Many big companies in Indonesia produce potato chips. One of them is Istana Factory located in Batu Tourism City. The potatoes processed have various sizes from the largest size (Super), medium size (AB), and then the smallest size (A). The process of sorting at the factory has been done by human manually, then it will produce less relative output. Therefore, it is needed a research about a tool that can sort the potato chips automatically. In this study, the system made is in the form of conveyor, which a webcam is installed as a censor to take pictures from potato chips, then those are processed in Raspberry Pi using image processing of Background Subtraction method. Background Subtraction is a method widely used to detect moving objects which chip taken its value in this system is a moving object. Potato chips will be classified based on size read by the system by using the  $w$  and  $h$  value parameters or the width and height of the bounding box of potato chips that are converted to actual size by millimeters. The value is used as a reference to be classified with the Naïve Bayes method. Naïve Bayes is used for the classification method because it is a method that has high performance and has excellent accuracy for classification. After getting the result of classification, potato chips will be sorted by servo-driven impeller. From the results of test conducted Background Subtraction can read the image of potatoes well. The reading of potato chip size from the system gets a small error of 3.73%. Then the accuracy obtained with Naïve Bayes method in chips classification with 90 training data and 30 test data is worth 93.33% having an average processing time speed of 1.7 ms from 30 times of the test. Then it is performed a test of hardware servo that has been running based on the system.

**Keywords:** Potato chips, Background Subtraction, classification, Naive Bayes.

## DAFTAR ISI

LEMBAR PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i> .....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Tinjauan Pustaka .....	5
2.2 Dasar Teori.....	7
2.2.1 Keripik Kentang Istana Batu .....	7
2.2.2 Citra Digital.....	7
2.2.3 Jenis Citra .....	8
2.2.4 Background Substraction .....	8
2.2.5 Rasperry Pi 3 .....	9
2.2.6 Ardunio Uno (Atmega328P).....	10
2.2.7 Webcam .....	12
2.2.8 OpenCV .....	12
2.2.9 Motor Stepper NEMA-17 .....	13
2.2.10 A4988 Driver Motor .....	15
2.2.11 Motor Servo .....	16
2.2.12 Naïve Bayes .....	17

BAB 3 METODOLOGI .....	19
3.1 Studi Literatur .....	20
3.2 Analisis Kebutuhan .....	20
3.3 Pengumpulan Data .....	21
3.4 Perancangan Sistem.....	21
3.4.1 Perancangan Perangkat Keras .....	21
3.4.2 Perancangan Perangkat Lunak.....	21
3.5 Implementasi .....	22
3.6 Pengujian dan Analisis.....	22
3.7 Kesimpulan.....	23
BAB 4 REKAYASA PERSYARATAN .....	24
4.1 Gambaran Umum Sistem.....	24
4.1.1 Tujuan.....	24
4.1.2 Karakteristik Pengguna .....	24
4.1.3 Lingkungan Operasi.....	24
4.1.4 Batasan Perancangan dan Implementasi.....	24
4.1.5 Asumsi dan Ketergantungan.....	25
4.2 Analisis Kebutuhan Sistem.....	25
4.2.1 Kebutuhan Fungsional.....	25
4.2.2 Kebutuhan Non Fungsional.....	26
4.3 Pengumpulan Data .....	27
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	28
5.1 Perancangan Sistem.....	28
5.1.1 Perancangan <i>Prototype</i> Alat Pemilah Keripik Kentang.....	28
5.1.2 Perancangan Perangkat Keras .....	29
5.1.3 Perancangan Perangkat Lunak.....	32
5.2 Implementasi Sistem .....	42
5.2.1 Implementasi <i>Prototype</i> Alat Pemilah Kentang.....	42
5.2.2 Implementasi Perangkat Keras .....	43
5.2.3 Implementasi Perangkat Lunak.....	45
BAB 6 PENGUJIAN DAN ANALISIS.....	56
6.1 Pengujian Pengambilan Nilai Objek Keripik Kentang Menggunakan <i>Background Subtraction</i> .....	56



6.1.1 Prosedur pengujian .....	56
6.1.2 Hasil dan Analisis Pengujian .....	56
6.2 Pengujian Pengambilan Gambar Keripik Kentang dari Sistem .....	57
6.2.1 Prosedur Pengujian .....	58
6.2.2 Hasil dan Analisis Pengujian .....	58
6.3 Pengujian Akurasi Hasil Klasifikasi <i>Naive Bayes</i> .....	60
6.3.1 Tujuan Pegujian .....	60
6.3.2 Prosedur Pegujian .....	60
6.3.3 Hasil dan Analisis Pengujian .....	60
6.4 Pengujian Servo Pendorong Keripik Kentang .....	62
6.4.1 Tujuan Pegujian .....	62
6.4.2 Prosedur Pengujian .....	62
6.4.3 Hasil dan Analisis Pengujian .....	63
6.5 Pengujian Waktu Komputasi Klasifikasi Pemrosesan Citra .....	64
6.5.1 Tujuan Pegujian .....	64
6.5.2 Prosedur Pegujian .....	64
6.5.3 Hasil dan Analisis .....	64
BAB 7 PENUTUP .....	67
7.1 Kesimpulan.....	67
7.2 Saran .....	67
DAFTAR PUSTAKA.....	69
LEMPIRAN A DATA LATIH .....	71
LAMPIRAN B KODE PROGRAM.....	73
B.1 Kode Program Raspberry Pi.....	73
B.2 Kode Program Arduino .....	76

## DAFTAR GAMBAR

Gambar 2.1 Keripik kentang Istana .....	7
Gambar 2.2 Background Substraction .....	9
Gambar 2.3. Raspberry Pi 3.....	10
Gambar 2. 4 GPIO Raspberry Pi 3 .....	10
Gambar 2.5 Arduino Uno .....	11
Gambar 2.6 Webcam .....	12
Gambar 2. 7 Logo OpenCV .....	13
Gambar 2.8 Prinsip Kerja Motor Stepper.....	13
Gambar 2.10 Driver Motor A4988 .....	15
Gambar 2.11 Motor Servo MG995 .....	16
Gambar 3. 1 Diagram Alir Metodologi Penelitian.....	19
Gambar 3.2 Perancangan perangkat keras.....	21
Gambar 3.3 Perancangan perangkat lunak.....	22
Gambar 5.1 Desain <i>Prototype</i> Alat Pemilah Kentang Otomatis .....	28
Gambar 5.2 Skematik Rangkaian Sistem.....	29
Gambar 5.3 Diagram Alir Program Utama Sistem .....	33
Gambar 5.4 Diagram Alir Perancangan Program Stepper .....	33
Gambar 5.5 Diagram alir perancangan perangkat lunak pengambilan data dari citra .....	34
Gambar 5.6 Diagram alir perancangan klasifikasi <i>Naïve Bayes</i> .....	35
Gambar 5.7 Diagram alir ProbPrior.....	36
Gambar 5.8 Diagram alir fungsi Gaussian() .....	37
Gambar 5.9 Diagram alir ProbPosterior() .....	38
Gambar 5.10 Diagram alir perancangan servo .....	42
Gambar 5.11 Implementasi Prototype Alat Pemilah Kentang.....	43
Gambar 5.12 Implementasi rangkaian stepper dan driver dengan Arduino Uno	43
Gambar 5.13 Implementasi rangkaian Raspberry Pi dan Webcam serta LED .....	44
Gambar 5.14 Implementasi rangkaian Raspberry Pi dan arduino Uno .....	44
Gambar 5.15 Implementasi rangkaian Arduino dengan servo .....	44
Gambar 6.1 Gambar frame yang diambil.....	57
Gambar 6.2 Gambar Biner frame Background Subtraction dan Opening.....	57

Gambar 6.3 Screenshoot data yang diambil dari bounding box keripik kentang	57
Gambar 6.4 Hasil pengujian waktu komputasi .....	665



## DAFTAR TABEL

Tabel 2.1 Perbandingan Tinjauan Pustaka .....	6
Tabel 2.2 Spesifikasi Raspberry Pi 3 .....	9
Tabel 2.3 <i>PinOut</i> Arduino Uno Atmega328.....	11
Tabel 2.4 Spesifikasi Stepper Motor Nema 17 .....	14
Tabel 2.5 Spesifikasi servo MG995.....	16
Tabel 5.1 Keterangan Koneksi Driver Motor dengan Power Supply.....	29
Tabel 5.2 Keterangan Koneksi Arduino dengan Power Supply.....	30
Tabel 5.3 Keterangan Koneksi Stepper dengan Driver Motor .....	30
Tabel 5.4 Keterangan Koneksi Driver Motor dengan Arduino Uno .....	30
Tabel 5.5 Keterangan Koneksi Raspberry dan Arduino Uno.....	30
Tabel 5.6 Keterangan koneksi pin servo dengan power supply .....	31
Tabel 5.7 Keterangan koneksi pin Arduino Uno dengan Servo .....	31
Tabel 5.8 Keterangan koneksi Webcam dengan Raspberry Pi 3.....	31
Tabel 5.9 <i>Mean</i> tiap jenis keripik kentang .....	39
Tabel 5.10 Tabel standart deviasi tiap jenis keripik kentang.....	40
Tabel 5.11 Tabel gaussian tiap jenis keripik kentang.....	41
Tabel 5.12 Kode pemrograman inisialisasi library Sistem Pemilah Kentang pada Raspberry .....	45
Tabel 5.13 Pengambilan gambar video.....	45
Tabel 5.14 Kode pemrograman inisialisasi library Sistem Pemilah Kentang pada Arduino Uno .....	46
Tabel 5.15 Kode program stepper motor .....	46
Tabel 5.16 Proses pengambilan data pada citra keripik kentang .....	47
Tabel 5.17 Pengambilan data uji keripik kentang.....	50
Tabel 5.18 Kode program inisialisasi variabel untuk klasifikasi Naive Bayes.....	50
Tabel 5.19 Kode Program Fungsi Gaussian .....	51
Tabel 5.20 Kode program fungsi posterior .....	52
Tabel 5.21 Kode program fungsi kesimpulan.....	52
Tabel 5.22 Kode program servo .....	53
Tabel 6.1 Hasil pengujian pembacaan kentang dari sistem.....	59
Tabel 6.2 Data uji dan hasil pengujian <i>Naive Bayes</i> .....	60

Tabel 6.3 Hasil pengujian servo pendorong keripik kentang.....	63
Tabel 6.4 Kode program fungsi “milis()” .....	64
Tabel 6.5 Pengujian Waktu Komputasi .....	64





## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Perkembangan ilmu pengetahuan dan teknologi pada masa sekarang banyak digunakan di dunia industri untuk pengolahan hasil pertanian dan perkebunan. Salah satu pengolahan pada pengolahan hasil pertanian dan perkebunan adalah pemilihan produk berdasarkan kualitas, seperti tingkat kematangan berdasarkan warna, berat atau ukuran. Penggunaan tenaga manusia (manual) untuk menentukan atau mengklasifikasi tingkat kelas sebuah objek berdasarkan ukuran atau parameter lainnya memiliki beberapa kekurangan antara lain penilaian manusia yang bersifat subyektif dan tidak konsisten terhadap objek buah serta pekerjaan yang dilakukan secara berulang-ulang dapat menyebabkan kejenuhan (Arga Anugrahandy, 2013). Proses penyortiran pada suatu pabrik/perusahaan umumnya dilakukan oleh manusia secara manual sehingga menghasilkan keluaran yang relatif kurang seragam. Hal itu disebabkan oleh faktor kurang teliti, kelelahan, salah persepsi dan lain-lain yang menyebabkan seseorang melakukan kesalahan saat penyortiran karena beragamnya visual yang dimiliki setiap manusia.

Kentang adalah jenis sayuran yang bisa diolah menjadi beragam makanan, salah satunya ialah keripik. Banyak perusahaan besar di Indonesia yang memproduksi keripik dari kentang. Salah satunya yaitu Pabrik Istana yang berada di Kota Wisata Batu. Kentang yang diolah mempunyai beragam ukuran mulai dari ukuran yang paling besar yaitu (Super), lalu ada ukuran AB yang berukuran sedang, kemudian ukuran A yang paling kecil. Dari setiap ukuran tersebut akan diolah menjadi keripik berdasarkan kelasnya masing-masing (Devi, 2018). Pengolahan keripik dari kentang tersebut sejauh ini masih disortir atau diklasifikasi oleh manusia sehingga belum ada keseragaman dari keripik kentang tersebut, oleh sebab itu dibutuhkan suatu sistem yang dapat menyortir secara otomatis agar lebih seragam yang sesuai dengan kelasnya masing-masing. Proses pemilihan secara otomatis bertujuan untuk menentukan kelas sejenis dari suatu objek.

Background subtraction adalah suatu metode pengambilan data suatu gambar yang banyak digunakan pada proyek-proyek berbasis pengolahan citra yang penempatan kameranya adalah statis atau tidak bergerak. Salah satu aplikasi yang sering menggunakan fungsi dari Background Subtraction ini adalah aplikasi penghitung jumlah pengunjung yang memasuki maupun meninggalkan ruangan ataupun kendaraan yang melewati suatu jalur dalam sistem informasi lalu lintas dan kemudian mengklasifikasikannya berdasarkan golongan kendaraan. Metode ini memisahkan antara objek yang bergerak dengan *foreground* dan *background*. Seperti penelitian yang berjudul *Penerapan Algoritma Background Subtraction Untuk Tracking dan Klasifikasi Kendaraan* yang dilakukan oleh Muhammad Ikhsan Samir, ia meneliti objek bergerak dan di jalan raya kemudian mengklasifikasi kendaraan berdasarkan golongan (Samir, n.d.).

Dalam pengklasifikasian keripik kentang sistem membutuhkan nilai-nilai panjang ( $wmm$ ) dan lebar ( $hmm$ ) dalam satuan milimeter. Selain itu sistem juga harus dapat menghitung hingga dapat mengklasifikasinya secara akurat. Sistem pemilah keripik kentang ini menggunakan metode Naive Bayes dalam pengklasifikasiannya, dikarenakan Naive Bayes merupakan salah satu metode pembelajaran supervised yang mudah, efisien, efektif dan handal menangani derau data seperti atribut yang tidak relevan (Izzati Saila Hafsa, 2015).

Berdasarkan kebutuhan akan pengklasifikasian dan penyortiran keripik kentang agar seragam saat produksi keripik, maka dibutuhkan alat yang secara otomatis memilah keripik kentang berdasarkan ukurannya sehingga dapat mempermudah dan meringankan pekerjaan manusia. Alat didesain untuk memasukkan keripik kentang pada kelasnya masing-masing dengan menggunakan Arduino, mini PC Raspberry pi dan webcam untuk pengolahan citra untuk mendapatkan hasil yang lebih konsisten daripada pengerjaan dari tenaga manusia.

## 1.2 Rumusan masalah

Setelah dibuat uraian mengenai latar belakang diatas, maka dirumuskan masalah pada penelitian ini sebagai berikut :

1. Apakah *Background Subtraction* dapat membaca nilai *pixel* dari keripik kentang untuk diambil nilai panjang dan lebar?
2. Berapakah tingkat keakuratan mengklasifikasi keripik kentang menggunakan *Naïve Bayes* ?
3. Bagaimana hardware pada aktuator bekerja dari implementasi *Background Subtraction* untuk Klasifikasi Keripik Kentang Berbasis Raspberry Pi menggunakan Metode *Naïve Bayes*?
4. Bagaimana waktu komputasi dari performansi Implementasi *Background Subtraction* untuk klasifikasi keripik kentang berbasis Raspberry Pi?

## 1.3 Tujuan

Tujuan dilakukan penelitian pada sistem ini adalah sebagai berikut :

1. Melakukan perancangan alat yang dapat mengklasifikasi keripik kentang secara otomatis
2. Membaca data dari kentang untuk diklasifikasikan
3. Menerapkan metode *Naïve Bayes* untuk klasifikasi untuk mengetahui tingkat keakuratan dari sistem
4. Mengetahui peformansi alat dari implementasi *Background Subtraction* untuk Klasifikasi Keripik Kentang Berbasis Raspberry Pi menggunakan Metode *Naïve Bayes*

## 1.4 Manfaat

Manfaat dari penelitian ini adalah :

1. Memudahkan, meringankan pekerjaan manusia
2. Meminimalisir kesalahan dalam pengklasifikasian keripik kentang

### 1.5 Batasan masalah

1. Objek yang diteliti adalah keripik kentang dari perusahaan keripik Istana Kota Batu
2. Proses pengambilan gambar dilakukan di dalam ruangan (*indoor*).
3. *Background* dari keripik kentang berwarna biru polos.
4. Proses pengolahan citra menggunakan perangkat lunak opencv.
5. Menggunakan Raspberry Pi sebagai pengolah data
6. Menggunakan *Naïve Bayes* untuk klasifikasi.
7. Tinggi dari kamera sudah ditetapkan atau tidak diubah-ubah
8. Kecepatan dari belt harus stabil atau tetap

### 1.6 Sistematika pembahasan

Pada sistematika pembahasan akan diuraikan penjelasan secara singkat masing-masing bab sebagai berikut:

#### **BAB I : Pendahuluan**

Bab pendahuluan membahas tentang latar belakang atau pokok masalah yang akan dibahas dari sistem pemilah keripik kentang, menentukan rumusan masalah, tujuan, manfaat, metode yang dipakai untuk menyelesaikan masalah secara keseluruhan.

#### **BAB II : Landasan Kepustakaan**

Bab ini menjelaskan tentang dasar teori yang digunakan sebagai referensi dalam penyusunan skripsi, lalu membandingkan penelitian yang akan dibuat dengan penelitian terdahulu.

#### **BAB III : Metode Penelitian**

Pada bab ini menjelaskan alur kerja sistem yang dilakukan dalam penulisan skripsi meliputi studi literatur, analisis kebutuhan, perancangan, implementasi dan pengujian sistem yang akan dibuat.

#### **BAB IV : Rekayasa Persyaratan**

Pada bab rekayasa persyaratan menjelaskan secara terperinci tentang gambaran umum dari sistem, kebutuhan sistem, kebutuhan fungsional dan non fungsional dari sistem dan pengumpulan data.

#### **BAB V : Perancangan dan Implementasi**

Pada bab perancangan dan implementasi menjelaskan menjelaskan perancangan sistem yang dibuat berdasarkan diagram blok serta hasil implementasi perancangan sistem pemilah keripik kentang.

**BAB VI : Pengujian**

Pada bab pengujian membahas tentang pengujian sistem telah dibuat. Pengujian dilakukan untuk melakukan uji coba sistem pemilah keripik kentang sehingga diketahui seberapa kesuksesan sistem bekerja.

**BAB VII Penutup**

Pada bab ini berisi tentang saran dan kesimpulan dari sistem yang telah dibuat dan disampaikan saran-saran yang dapat digunakan untuk penelitian selanjutnya.

**BAB VII Penutup**

Pada bab ini berisi tentang saran dan kesimpulan dari sistem yang telah dibuat dan disampaikan saran-saran yang dapat digunakan untuk penelitian selanjutnya.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Tinjauan Pustaka

Pada penelitian “Sistem Pendeteksi Gerak Berbasis Web Menggunakan Metode *Background Subtraction*” yang dilakukan Dias Prohatmoko, penulis membuat suatu sistem pemantau rumah menggunakan web yang dilengkapi dengan fitur pendeteksian gerak. Kamera hanya akan mendeteksi dan merekam apabila ada gerak. Hal ini akan meningkatkan efisiensi penggunaan kamera. Studi ini digunakan untuk pemantauan suatu rumah dan gedung. Sistem pemantau mempunyai kemampuan untuk mendeteksi pergerakan suatu objek. Metode deteksi gerak yang digunakan adalah teknik pengambilan gambar *background subtraction* (Dias Prihatmoko, 2015).

Pada penelitian “Implementasi Sistem Deteksi Slot Parkir Mobil Menggunakan Metode *Morfologi* dan *Background Subtraction*” yang dilakukan oleh Maulana, penulis menggunakan sebuah sistem *embedded* untuk mendeteksi ketersediaan slot parkir menggunakan modul kamera Raspberry Pi V2 dan Raspberry Pi 3. Pada sistem juga diterapkan metode morfologi dan *background subtraction* sehingga sistem dapat mendeteksi jumlah slot parkir yang ada sesuai dengan keadaan sebenarnya sehingga pengendara tidak perlu membuang banyak waktu hanya untuk sekedar mencari tempat parkir. (Maulana, 2018).

Pada penelitian “Otomasi Pemisah Buah Tomat Berdasarkan Ukuran Menggunakan Webcam Sebagai Sensor” yang dilakukan oleh Thiang, penulis membuat suatu sistem yang dapat mengenali tingkat kelayakan pemasaran buah tomat yang memanfaatkan citra luas permukaan ukuran dari gambar buah tomat yang diambil. Sistem ini berfungsi untuk membedakan buah tomat secara digital sehingga dapat membedakan ukuran dari buah tomat yang akan dipisahkan dan diharapkan menghasilkan akurasi yang tinggi (Thiang.,2008).

Berdasarkan pada beberapa penelitian diatas, penulis tertarik untuk mengembangkan dari penelitian ketiga yang mengklasifikasi buah berdasarkan ukuran, tetapi dengan objek berbeda serta pengolahan gambar dan metode pengklasifikasian yang berbeda. Yaitu dengan mengklasifikasi jenis keripik kentang berdasarkan ukuran menggunakan metode pengolahan gambar *Background Subtraction* dan menggunakan metode *Naïve Bayes* untuk klasifikasi.



Adapun perbandingan tinjauan pustaka dengan yang penelitian yang diusulkan terdapat pada Tabel 2.1 dibawah ini.

**Tabel 2.1 Perbandingan Tinjauan Pustaka**

No	Nama Penulis [Tahun], Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	Dias Prohatmoko [2015], Sistem Pendeteksi Gerak Berbasis Web Menggunakan Metode <i>Background Substraction</i>	Menggunakan metode pengolahan gambar yang sama menggunakan metode <i>Background Substraction</i> untuk mendeteksi objek, dan menggunakan webcam untuk mengambil gambar.	Objek yang digunakan adalah gerakan dari manusia atau apapun yang bergerak diruangan	Objek yang digunakan adalah keripik kentang
			Tidak menggunakan metode hanya mendeteksi, dan tidak menggunakan perangkat mikrokontroller, karena tidak diklasifikasi	Menggunakan metode klasifikasi <i>Naïve Bayes</i> , dan menggunakan perangkat <i>embedded</i> .
2	Maulana[2018], Implementasi Sistem Deteksi Slot Parkir Mobil Menggunakan Metode <i>Morfologi</i> dan <i>Background Substraction</i>	Menggunakan metode pengolahan gambar yang sama menggunakan <i>Background Substraction</i> , dan <i>morfologi</i> untuk menghilangkan noise serta menggunakan perangkat <i>embedded</i>	Mendeteksi objek slot parkir mobil	Mendeteksi objek keripik kentang
			Menggunakan modul kamera V2 dari Raspberry	Menggunakan Webcam

3	Thiang.[2008], Otomasi Pemisah Buah Tomat Berdasarkan Ukuran Menggunakan Webcam Sebagai Sensor	Klasifikasi objek berdasarkan ukuran dan menggunakan konveyor	Menggunakan edge detection untuk pengolahan gambar	Menggunakan Background Subtraction untuk pengolahan gambar
			Tidak menggunakan metode pengklasifikasian	Menggunakan Metode Naïve Bayes untuk pengklasifikasian

## 2.2 Dasar Teori

### 2.2.1 Keripik Kentang Istana Batu

Keripik kentang adalah salah satu cemilan yang banyak diminati oleh banyak orang. selain bergizi keripik ini dapat diolah dengan berbagai varian rasa sesuai keinginan. Pabrik keripik kentang Istana adalah suatu perusahaan yang menjual aneka pengolahan keripik kentang, yang memproduksi suatu keripik dengan beragam jenis yang dapat dilihat dari ukurannya yaitu ukuran SUPER, AB dan A. Dari masing-masing ukuran tersebut memiliki harga yang berbeda. Selain menjual keripik yang sudah jadi, perusahaan keripik kentang istana juga menjual keripik mentah yang belum diolah. Pengolahan keripik kentang dapat dilihat pada Gambar 2.1.



**Gambar 2.1 Keripik kentang Istana**

(Sumber : Keripik Kentang Istana)

### 2.2.2 Citra Digital

Citra digital adalah suatu gambar dua dimensi yang dapat ditampilkan pada komputer sebagai suatu himpunan atau diskrit nilai digital yang disebut dengan *pixel*. Pikel adalah kumpulan dari ribuan titik yang sangat kecil dan tiap-tiap titik tersebut memiliki warna tertentu. Citra digital didefinisikan pada variabel  $f(x,y)$ ,

yang merupakan intensitas cahaya pada citra koordinat tersebut, dimana  $x$  dan  $y$  adalah koordinat spasial. Fungsi tersebut di setiap titik  $(x,y)$  merupakan tingkat kecemerlangan citra pada titik tersebut.

### 2.2.3 Jenis Citra

Berikut adalah jenis-jenis citra berdasarkan nilai *pixel*nya:

#### 1. Citra Biner

Citra biner adalah citra digital memiliki dua kemungkinan nilai *Pixel* yaitu hitam(0) dan putih(1), yang disebut juga dengan citra B &W (*black and white*) atau monokrom. 1 bit digunakan untuk mewakili nilai setiap *Pixel* dari citra biner. Citra biner biasanya dihasilkan dari proses pengolahan seperti segmentasi, pengembangan ataupun morfologi.

#### 2. Citra Grayscale

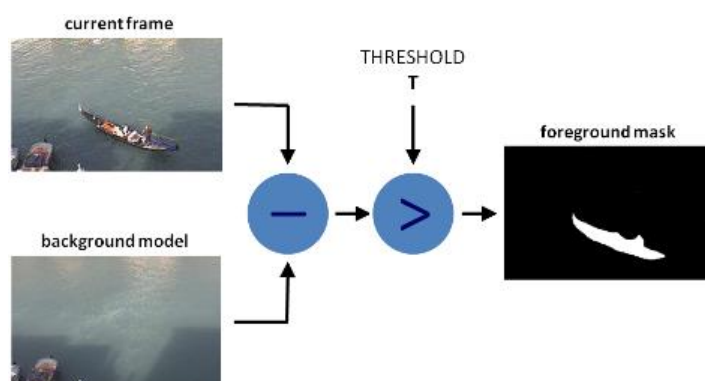
Citra grayscale adalah pada setiap *pixel* merepresentasi derajat intensitas cahaya atau keabuan. Terdapat 256 jenis derajat keabuan pada citra grayscale, mulai dari putih, hingga semakin gelap sampai warna hitam. Oleh karena terdapat kemungkinan 256 derajat keabuan, maka setiap *Pixel* pada grayscale disimpan 1 byte memory (8 bits).

#### 3. Citra Warna

Citra warna terdiri dari 3 layar *matrix*, yaitu *R-layer*, *G-layer*, *B-layer*. Sistem warna RGB (Red, Green, Blue) menggunakan sistem tampilan grafik kualitas tinggi yaitu 24 bit. Setiap komponen warna merah, hijau, biru masing-masing mendapatkan alokasi 8 bit untuk menampilkan warna.

### 2.2.4 Background Substraction

Background Substraction adalah proses untuk menemukan objek pada gambar dengan cara membandingkan suatu gambar yang ada dengan sebuah model latar belakang. *Background Substraction* merupakan salah satu teknik bidang citra dan komputer vision dengan cara mengambil atau mendeteksi *foreground* dari *background* untuk diproses lebih lanjut seperti pengenalan objek dll. Pada umumnya Background Substraction digunakan untuk mendeteksi objek bergerak seperti mendeteksi objek manusia, teks, kendaraan, dll. Gambar 2.2 merupakan contoh pengolahan gambar menggunakan background subtraction.



**Gambar 2.2** Background Substraction

(Sumber : <https://docs.opencv.org>)

### 2.2.5 Raspberry Pi 3

Raspberry Pi merupakan sebuah mini PC atau modul mikro komputer. Raspberry Pi 3 adalah generasi ketiga dari keluarga Raspberry Pi. Raspberry mempunyai sebuah input output digital port seperti pada board mikrokontroller. Raspberry Pi 3 mempunyai 1GB RAM dan grafis Broadcom VideoCore IV pada frekuensi clock yang lebih tinggi dan lebih baik dari Raspberry generasi sebelumnya yang berjalan pada 250 MHz. Spesifikasi dari dituliskan pada Tabel 2.2:

**Tabel 2.2** Spesifikasi Raspberry Pi 3

<b>SoC</b>	Broadcom BCM2837
<b>CPU</b>	4x ARM Cortex-A53, 1.2GHz
<b>GPU</b>	Broadcom VideoCore IV
<b>RAM</b>	1GB LPDDR2 (900 MHz)
<b>Networking</b>	10/100 Ethernet, 2.4GHz 802.11n wireless
<b>Bluetooth</b>	Bluetooth 4.1 Classic, Bluetooth Low Energy
<b>Storage</b>	microSD
<b>GPIO</b>	40-pin header, populated
<b>Ports</b>	HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

Mini PC Raspberry Pi 3 ditunjukkan pada Gambar 2.3 :

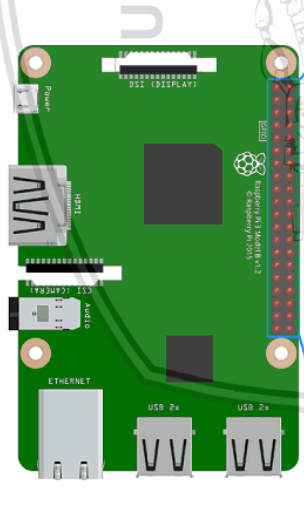


**Gambar 2.3.** Raspberry Pi 3

Sumber : (Sumber: [www.raspberrypi.org](http://www.raspberrypi.org))

### GPIO Raspberry Pi 3

*General-Purpose Input/Output* (GPIO) merupakan pin generik pada sirkuit terpadu yang digunakan untuk input atau output dan dapat dikontrol atau diprogram oleh pengguna saat dijalankan. Pada Gambar 2. 4 adalah pin GPIO Raspberry Pi 3.



3.3V	1	2	5V
GPIO2 (SDA1)	3	4	5V
GPIO3 (SCL1)	5	6	GND
GPIO4 (GPIO_GCLK)	7	8	GPIO14 (UART_TXD0)
GND	9	10	GPIO15 (UART_RXD0)
GPIO17 (GPIO_GEN0)	11	12	GPIO18 (GPIO_GEN1)
GPIO27 (GPIO_GEN2)	13	14	GND
GPIO22 (GPIO_GEN3)	15	16	GPIO23 (GPIO_GEN4)
3.3V	17	18	GPIO24 (GPIO_GEN5)
GPIO10 (SPI0_MOSI)	19	20	GND
GPIO9 (SPI0_MISO)	21	22	GPIO25 (GPIO_GEN6)
GPIO11 (SPI0_CLK)	23	24	GPIO8 (SPI_CE0_N)
GND	25	26	GPIO7 (SPI_CE1_N)
ID_SD (I2C EEPROM)	27	28	ID_SC (I2C EEPROM)
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

**Gambar 2. 4** GPIO Raspberry Pi 3

(Sumber : <http://www.electronicwings.com>)

### 2.2.6 Arduino Uno (Atmega328P)

Arduino Uno adalah sebuah board mikrokontroler ATmega328, yang memiliki pin sebanyak 14 yang merupakan digital *input/output*, 6 diantaranya adalah output *Pulse Width Modulation*(PWM)), 6 pin analog, sebuah 16 MHz kristal osilator, sebuah *power jack*, sebuah *ICSP header*, sebuah *port* USB, dan tombol *reset* (Arduino, 2016).



Arduino Uno dapat mudah dihubungkan ke sebuah komputer yaitu dengan menggunakan kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk catudaya. Arduino mempunyai semua yang dibutuhkan untuk melakukan sebuah kontrol. Pada Gamabr2.5 adalah bentuk dari mikrokontroller Arduino uno.



**Gambar 2.5** Arduino Uno

Sumber: (Arduino, 2016)

Pada Tabel 2.3 spesifikasi dari Arduino Uno Atmega328 dan deskripsi dari masing-masing pin :

**Tabel 2.3** *PinOut* Arduino Uno Atmega328

<b>Mikrontrroller</b>	<b>ATmega328P</b>
Tegangan Opeasi	5V
Tegangan Input (direkomendasikan)	7-12V
Tegangan <i>Input</i> (batas)	6-20V
Pin I/O Digital	14 (6 diantaranya menunjang output PWM)
Pin I/O PMW Digital	6
Pin Analog <i>Input</i>	6
Arus DC tiap Pin I/O	20 mA
Arus DC untuk Pin 3.3V	50 mA
Memori <i>Flash</i>	32 KB (ATmega328P) Dimana 0,5 Kb digunakan oleh boatloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Kecepatan <i>Clock</i>	16 MHz
Panjang	68.6 mm
Lebar	53.4 mm
Berat	25 g

Sumber: (Arduino, 2016)

### 2.2.7 Webcam

Webcam adalah suatu kamera yang dapat dihubungkan ke PC yang digunakan untuk berbagai macam kebutuhan yang dihubungkan melalui port USB. Webcam atau Web camera adalah kamera yang gambarnya bisa di akses menggunakan aplikasi yang dapat menampilkan video menggunakan PC, dan juga dapat ditampilkan *world wide web* (www) atau program instant messaging.

Webcam sangat bermanfaat pada berbagai macam bidang seperti bidang telekomunikasi, industri, keamanan. Sebagai contoh dimanfaatkan untuk monitoring suatu objek, sistem keamanan, video call, sebagai sensor dan lain-lain.

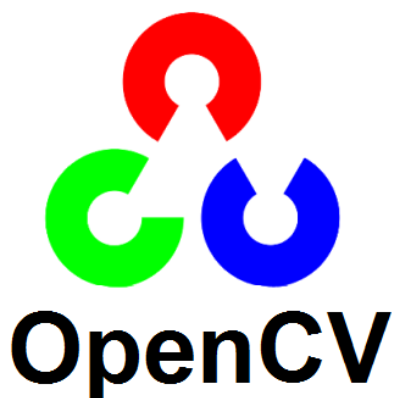


**Gambar 2.6 Webcam**  
(Sumber : [www.mikronis.hr](http://www.mikronis.hr))

### 2.2.8 OpenCV

OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) pustaka yang digunakan untuk pengolahan citra computer vision. Komputer Vision adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (Image Processing) yang memungkinkan komputer dapat melihat seperti manusia. Dengan vision tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari Computer Vision adalah Face Recognition, Face Detection, Face/Object Tracking, Road Tracking, dll.

OpenCV adalah pustaka Open Source Computer Vision untuk C, C++, Python. OpenCV didesain untuk aplikasi real-time, memiliki fungsi-fungsi akuisisi yang baik untuk image/video.



**Gambar 2. 7** Logo OpenCV

(Sumber : [www.opencv-srf.com](http://www.opencv-srf.com))

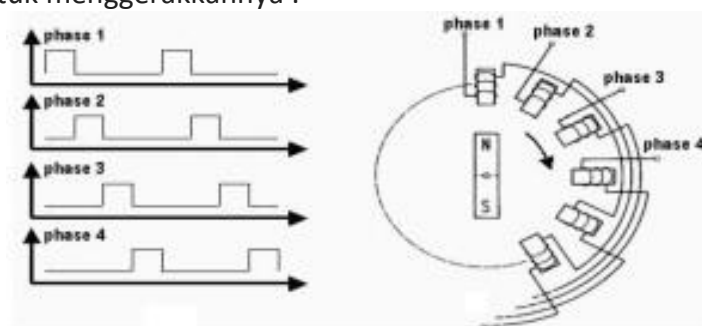
### 2.2.9 Motor Stepper NEMA-17

Motor stepper merupakan sebuah motor DC yang bergerak dalam langkah diskrit. Motor ini mempunyai sebuah fase atau beberapa gulungan pada motor. Motor akan berputar satu langkah atau step pada satu waktu jika diberi daya secara berturut-turut pada setiap tahap.

Motor stepper dikendalikan dengan pulsa digital. Prinsip kerja motor stepper adalah mengubah besaran elektronik menjadi gerakan besaran diskrit dimana motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor stepper tersebut(Zona,2015).

Stepper dapat dapat dikontrol atau dikendalikan atau diprogram menggunakan sebuah mikrokontroller, untuk mengatur langkah posisi serta kecepatan dari motor stepper. Oleh karenanya motor stepper banyak digunakan untuk sistem yang membutuhkan gerak yang presisi salah satunya untuk menggerakkan konveyor.

Pada Gambar 2.8 adalah ilustrasi struktur motor stepper dan sebuah pulsa yang dibutuhkan untuk menggerakkannya :



**Gambar 2.8** Prinsip Kerja Motor Stepper

Sumber : (Zona,2015)

Gambar diatas memberikan ilustrasi dari pulsa keluaran pengendali motor stepper dan penerapan pulsa tersebut pada motor stepper untuk menghasilkan arah putaran yang bersesuaian dengan pulsa kendali.

Motor stepper mempunyai beberapa keunggulan sebagai berikut:

- **Speed Control** – Motor stepper mempunyai kinerja sangat baik untuk mengatur kecepatan rotasi karena presisi dalam penambahan kecepatan dan control, stepper banyak digunakan untuk proses otomasi dan robotika.
- **Positioning** – Motor stepper mempunyai pergerakan dalam langkah yang tepat sesuai dengan posisi yang diinginkan, banyak digunakan untuk aplikasi yang memerlukan posisi yang tepat seperti 3D printer, CNC, dan lain-lain.
- **Torsi pada kecepatan rendah**– Motor stepper mempunyai kemampuan yang tidak dimiliki oleh motor DC yaitu tidak memiliki torsi yang kuat pada kecepatan rendah tapi sebaliknya motor stepper memiliki torsi maksimum pada kecepatan rendah, torsi kuat dengan presisi yang tinggi.

Motor stepper memiliki keterbatasan sbb:

- **Rendah Efisiensi** – Motor stepper cenderung lebih panas karena mengkonsumsi daya yang cukup besar meskipun tidak dibebani\
- **Torsi Terbatas pada Kecepatan tinggi** - Motor stepper akan berkurang torsi pada kecepatan tinggi dari pada saat kecepatan rendah. Beberapa stepper bisa dioptimalkan untuk kecepatan tinggi tetapi harus dipasangkan dengan *driver* khusus.
- **Tidak ada Feedback** – Motor stepper tidak mempunyai umpan balik untuk mengetahui posisi saat ini seperti servo, walaupun motor ini memiliki presisi tinggi untuk tujuan keselamatan *limit switch* atau detektor tetap diperlukan.

Step dan resolusi :

Stepper Mempunyai Jumlah langkah per revolusi berkisar dari 4 sampai 400. Jumlah langkah pada umumnya adalah 24, 48 dan 200. Derajat per langkah disebut dengan resolusi,  $1,8^\circ$  derajat per step maka motor stepper butuh 200 langkah untuk menghasilkan satu putaran penuh atau 1 revolusi. 1 resolusi =  $1,8^\circ$  derajat yang berarti 1 langkah dan 200 resolusi menghasilkan 200 langkah atau 1 revolusi, dalam satu revolusi mempunyai 360 derajat dalam 1x putaran.

**Tabel 2.4** Spesifikasi Stepper Motor Nema 17

Mosaic Part No.	STEPMOT-1
Manufacturer Part No.	42BYG228
Size	Nema 17
Drive System	Unipolar

Weight	0,24 Kg
Max continuous power	5 W
Rotor Intertia	22 g-cm <sup>2</sup>
Bearings	Ball
Leads	18 in. 26 AWG UL 1007
Insulation resistance	>100 M $\Omega$ at 500 VDC
Dielectric strength	500V/minute
Mounting hole space diagonal	1,73 in
Mounting screw	3mm dia. 0,5 mm pitch
Shaft diameter	0,197 in. (5mm)
Motor height	1,5 in
Motor footprint	1.7 in x 1,7 in
Ambient temperature	-10°C to +55°C

(Sumber : <http://www.mosaic-industries.com>)

#### 2.2.10 A4988 Driver Motor

Motor driver IC A4988 dipilih sebagai motor driver karena tidak boros pin arduino dan mudah cara dioperasikan. Motor driver pada umumnya hanya menaikkan tegangan output dari input arduino, hal ini sangat boros pin pada arduino mengingat tiap motor stepper bipolar memiliki 4 buah kabel. IC A4988 sangat mudah digunakan karena hanya terdapat dua buah input masukan yakni DIR dan STEP. DIR berfungsi untuk mengubah arah putaran searah jarum jam dan berlawanan jarum jam. STEP berfungsi untuk mengatur kecepatan motor stepper dengan memberikan sinyal *HIGH* dan *LOW* dengan jedanya. Gambar 2.9 merupakan gambar dari Driver Motor A4988.



**Gambar 2.9** Driver Motor A4988

(Sumber : [www.pololu.com](http://www.pololu.com))



### 2.2.11 Motor Servo

Motor servo mempunyai 2 type, yaitu type servo *standart* dan type servo *continuous*. Biasanya untuk tipe standart hanya dapat melakukan pergerakan sebesar  $180^\circ$ , sedangkan untuk tipe servo *continuous* dapat melakukan rotasi sebesar  $360^\circ$ . Pada dasarnya motor servo tersusun dari motor DC, rangkaian kontrol, *gear box*, dan potensiometer sebagaimana diperlihatkan pada gambar diatas. Motor servo adalah motor yang mampu bekerja dua arah CW (*Clock Wise*) dan CCW (*Counter Clock Wise*) dimana arah dan sudut pergerakan rotornya dapat dikendalikan dengan memberikan variasi lebar pulsa (*Duty Cycle*) sinyal PWM pada bagian pin kontrolnya.

Secara umum untuk mengakses motor servo tipe standart adalah dengan cara memberikan pulsa *high* selama 1,5 ms dan mengulangnya setiap 20 ms, maka posisi servo akan berada di tengah atau netral  $90^\circ$  untuk pulsa 1,25 ms, maka akan bergerak berkebalikan arah jarum jam dengan sudut  $0^\circ$ . Pulsa *high* selama 1,75 ms maka akan bergerak searah jarum jam sebesar  $180^\circ$ .

Spesifikasi Servo MG995 dituliskan pada Tabel 2.5 :

**Tabel 2.5 Spesifikasi servo MG995**

<i>Modulation</i>	<i>Modulation</i>
<i>Torque</i>	4.8V: 138.9 oz-in (10.00 kg-cm)
<i>Speed</i>	4.8V: 0.20 sec/60
<i>Weight</i>	1.94 oz (55.0 g)
<i>Dimensions</i>	<i>Length</i> :1.60 in (40.6 mm)
<i>Width</i>	0.78 in (19.8 mm)
<i>Height</i>	1.69 in (42.9 mm)
<i>Motor Type</i>	<i>Coreless</i>
<i>Gear Type</i>	<i>Metal</i>
<i>Rotation/Support</i>	<i>Dual Bearings</i>



**Gambar 2.10** Motor Servo MG995

(Sumber: <https://potentiallabs.com>)

### 2.2.12 Naïve Bayes

*Naive Bayes* adalah sebuah klasifikasi yang didasarkan pada aturan Bayes dan sekumpulan asumsi independensi kondisional, yang dimaksud independensi disini adalah tidak adanya ketergantungan antara tiap fitur dalam setiap kelas objek yang diklasifikasikan. Berikut merupakan rumus dari teorema *Bayes* (Barber, 2010) :

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad (2.1)$$

Keterangan dari **Pesamaan (2.1)** yakni :

- $P(y|x)$  : Peluang *posterior* (probabilitas kondisional) dari suatu hipotesis kelas  $y$  akan terjadi setelah diberikan data  $x$ .
- $P(x|y)$  : Peluang *likelihood* dari sebuah data  $x$  terjadi akan mempengaruhi hipotesis kelas  $y$ .
- $P(x)$  : Peluang *evidence*  $x$  terjadi tanpa memperhatikan hipotesis kelas/*evidence* lainnya, yakni jumlah total dari semua peluang *likelihood* yang dikalikan dengan peluang *prior*.
- $P(y)$  : Peluang *prior* (awal) hipotesis kelas  $y$  terjadi tanpa memperhatikan data yang diberikan.

Hipotesis dalam teorema *Bayes* merupakan label kelas yang menjadi target dalam sebuah klasifikasi, sedangkan *evidence* adalah fitur yang menjadi masukan dalam klasifikasi. *Naive Bayes* dilambangkan dengan  $P(X|Y)$ , dimana  $X$  adalah masukan yang berupa fitur-fitur dan  $Y$  adalah kelas dalam sebuah klasifikasi. Notasi  $P(X|Y)$  berarti peluang kelas  $Y$  didapatkan setelah fitur-fitur  $X$  diamati, notasi ini merupakan peluang *likelihood* dan  $P(Y)$  merupakan notasi dari peluang *prior*. Berikut ini adalah persamaan untuk rumus *Naive Bayes* (Barber, 2010):

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^q P(X_i|Y)}{P(X)} \quad (2.2)$$

Keterangan dari **Pesamaan (2.2)** yakni :

- $P(Y|X)$  : Peluang *posterior* (probabilitas kondisional) dari suatu kelas  $Y$  akan terjadi setelah mengamati fitur-fitur  $X$ .
- $\prod_{i=1}^q P(X_i|Y)$  : Peluang *likelihood* dari masing-masing fitur  $X$  terjadi akan mempengaruhi kelas  $Y$ .
- $P(Y)$  : Peluang *prior* (awal) hipotesis kelas  $Y$  terjadi tanpa memperhatikan data yang diberikan.
- $P(X)$  : Peluang *evidence*  $X$  terjadi tanpa memperhatikan kelas/*evidence* lainnya, yakni jumlah total dari semua peluang *likelihood* yang dikalikan dengan peluang *prior*.

Dalam perhitungan klasifikasi untuk setiap kelas  $Y$  yang berbeda akan mempunyai nilai  $P(X)$  yang sama, sehingga dalam penentuan klasifikasi Naive Bayes selanjutnya ditentukan dari nilai peluang terbesar antara tiap kelas  $Y$  dari hasil perhitungannya  $P(Y) \prod_{i=1}^q P(X_i|Y)$  (Astuti, 2016).

Beberapa permasalahan yang ada untuk menentukan nilai peluang dari suatu kondisi yang mudah adalah dengan menghitung peluang dari data diskrit. Namun dalam kenyataannya tidak semua data tersaji dalam bentuk diskrit, tetapi ada yang berbentuk data kontinyu. Untuk itu dalam melakukan proses klasifikasi terhadap data kontinyu dengan *Naive Bayes* terdapat 2 cara yakni (Astuti, 2016) :

1. Melakukan proses perubahan data kontinyu menjadi data diskrit (diskritisasi) terhadap setiap fitur yang akan diestimasi.
2. Menganggap setiap fitur sesuai dengan data latih menggunakan fungsi *univariate normal (Gaussian) distribution* yang ditunjukkan pada **Persamaan (2.3)**, dimana parameter utama dari fungsi *Gaussian* ini adalah *mean* ( $\mu$ ) dan *varian* ( $\sigma^2$ ).

$$P(X = x_i | Y = y_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \quad (2.3)$$

Parameter  $\mu_{ij}$  bisa didapatkan dari mean pada sampel  $X_i(\bar{x})$  dari semua data latih yang menjadi milik kelas  $y_i$ , sedangkan  $\sigma_{ij}^2$  dapat dipekirakan dari varian sampel ( $s^2$ ) dari data latih.

Adapun fungsi untuk mencari nilai *Mean* dapat dilihat pada **Persamaan (2.4)** berikut.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (2.4)$$

Perhitungan *mean* dilakukan dengan menjumlahkan seluruh nilai data suatu kelompok sampel, kemudian dibagi dengan jumlah sampel tersebut. Dimana  $\bar{x}$  merupakan rata-rata hitung,  $x_i$  merupakan nilai sampel ke- $i$ , dan  $n$  adalah jumlah sampel.

Adapun fungsi untuk mencari nilai standar deviasi dapat dilihat pada **Persamaan (2.5)** berikut.

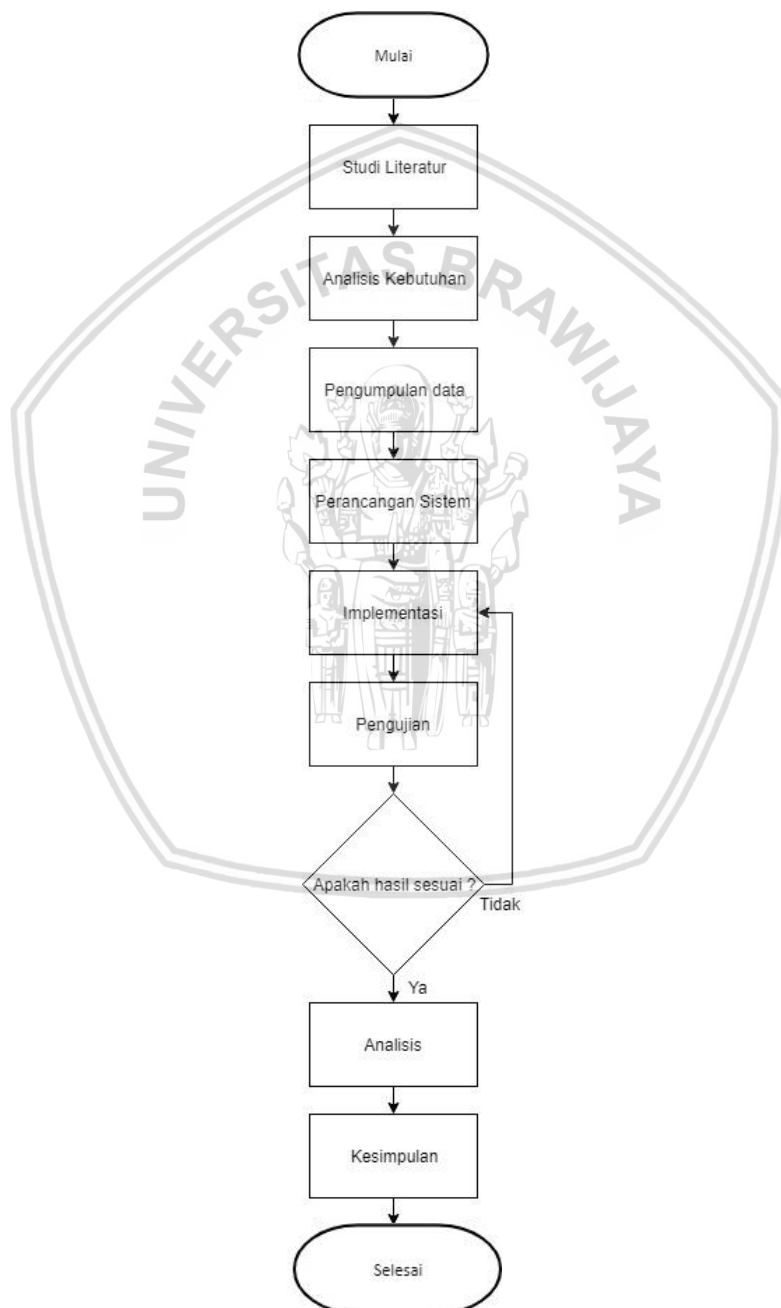
$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (2.5)$$

Untuk menghitung standar deviasi yaitu dengan mengurangi setiap nilai data dengan rata-rata kelompok data tersebut, selanjutnya semua hasil dijumlahkan kemudian dibagi dengan jumlah data secara keseluruhan dikurangi 1, dan terakhir hasilnya di akarkan. Dimana  $s$  merupakan standar deviasi (simpangan baku),  $x_i$  merupakan nilai  $x$  ke  $i$ ,  $\bar{x}$  merupakan rata-rata,  $n$  adalah ukuran sampel.

## BAB 3 METODOLOGI

Pada bab ini akan menjelaskan tahapan-tahapan yang digunakan dalam penyusunan skripsi, meliputi studi literatur, analisis kebutuhan sistem, pengumpulan data, perancangan sistem, implementasi sistem, pengujian, analisis, serta kesimpulan.

Alur metodologi penelitian yang dilakukan secara umum dapat diilustrasikan pada diagram alir Gambar 3.1:



**Gambar 3. 1 Diagram Alir Metodologi Penelitian**

### 3.1 Studi Literatur

Studi Literatur digunakan untuk landasan dasar dan penguat teori dalam menyusun penelitian. Teori pendukung diambil dari jurnal atau paper, buku, skripsi dan internet. Berikut dasar teori yang digunakan sebagai bahan studi :

1. Keripik Kentang Istana Batu
2. Citra Digital
3. Jenis Citra
4. *Background Substraction*
5. Raspberry Pi 3
6. Webcam
7. OpenCV
8. Motor Stepper
9. Motor Servo
10. *Naïve Bayes*

### 3.2 Analisis Kebutuhan

Tujuan dari analisis kebutuhan untuk menganalisis seluruh kebutuhan apasaja yang dibutuhkan untuk membuat sistem ini. Analisis kebutuhan dari sistem dilakukan dengan mengidentifikasi seluruh kebutuhan untuk membangun sistem, yang meliputi kebutuhan perangkat keras dan perangkat lunak.

#### 3.3.1 Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk membuat sistem pemilah keripik kentang ini adalah sebagai berikut:

- a. Raspberry Pi 3
- b. Laptop/Komputer
- c. Webcam
- d. Arduino Uno
- e. Motor stepper
- Motor Servo

#### 3.3.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk membuat sistem ini adalah sebagai berikut:

- a. *Operator Sistem (OS) Windows 10*
- b. *Image Processing Library*

- c. Bahasa pemrograman openCV

### 3.3 Pengumpulan Data

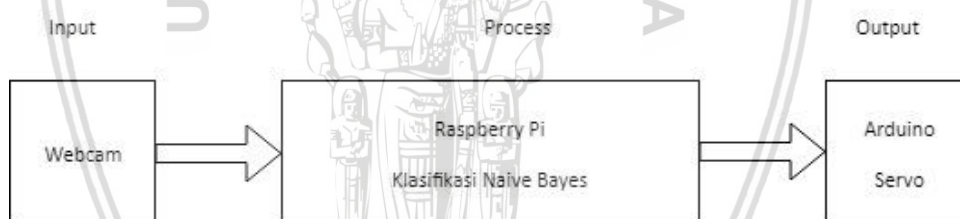
Pengumpulan data dilakukan dengan mengumpulkan data berupa keripik kentang istana dari Kota Batu. Waktu pelaksanaan pengambilan sampel dilakukan pada bulan Februari 2018. Data yang didapat akan digunakan sebagai data latih untuk basis pengklasifikasian serta sebagai data uji.

### 3.4 Perancangan Sistem

Perancangan dalam pembuatan sistem ini meliputi dua bagian yaitu perancangan perangkat keras dan perancangan perangkat lunak.

#### 3.4.1 Perancangan Perangkat Keras

Pada perancangan perangkat keras terdiri dari beberapa blok diagram dari sistem. Pada input menggunakan kamera webcam sebagai sensor, kentang berjalan diatas konveyor yang dijalankan dari stepper yang dikontrol dari arduino kemudian kamera menangkap citra yang melewati kamera dan dibantu pencahayaan dari led, dan outputnya adalah servo yang akan mengklasifikasikan kentang sesuai dengan jenisnya, apakah masuk golongan A, AB atau Super. Gambaran dari sistem dapat diilustrasikan pada Gambar 3.2:



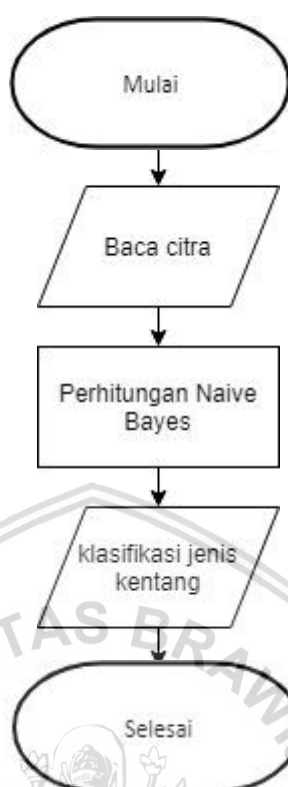
Gambar 3.2 Perancangan perangkat keras

Dalam sistem ini keripik masuk ke sebuah box untuk klasifikasi yang dijalankan oleh stepper motor pada belt konveyor, kemudian webcam mengambil gambar untuk diolah di Raspberry Pi menggunakan pengolahan gambar *Background Substravtion* dan diklasifikasi menggunakan metode *Naïve Bayes*, kemudian Raspberry mengirim perintah ke Arduino untuk memilah atau mendorong keripik kentang menggunakan servo sesuai kelasnya.

#### 3.4.2 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak berupa algoritma yang digunakan sesuai dengan tujuan utama dari pembuatan sistem yakni pembuatan program untuk pengklasifikasian keripik kentang dengan metode Naive Bayes. Algoritma perancangan perangkat lunak diilustrasikan pada Gambar 3.3 berikut:





**Gambar 3.3 Perancangan perangkat lunak**

Pada Gambar 3.3 dijelaskan bahwa sistem bekerja ketika ada sebuah inputan dari kamera masuk kemudian diklasifikasikan menggunakan metode *naïve bayes*.

### 3.5 Implementasi

Pada Implementasi sistem dilakukan implementasi berdasarkan perancangan sistem yang telah dibuat. Pada implementasi memberikan desain atau *prototype* sistem yang sudah dibuat dan beberapa potongan bahasa pemrograman yang digunakan dalam sistem.

### 3.6 Pengujian dan Analisis

Pengujian dan analisis yang diterapkan disini adalah dengan menguji akurasi data hasil pengklasifikasian keripik buah kentang sesuai kelasnya menggunakan *Background Substraction*. Ada beberapa skenario yang dilakukan untuk pengujian sistem yaitu :

1. Pengujian *Background Substraction*
2. Pengujian klasifikasi menggunakan *Naïve Bayes*
3. Pengujian hardware
4. Pengujian waktu komputasi

### 3.7 Kesimpulan

Pengambilan kesimpulan dibuat setelah melakukan semua proses pengujian selesai dan mendapatkan hasil yang akurat dan sesuai dengan database. Kesimpulan didapat dari hasil seluruh pengujian dan analisis terhadap sistem yang dilakukan.

Pada tahapan ini selain diambil kesimpulan, juga diharapkan ada saran untuk melakukan pengembangan lebih lanjut dan memperbaiki sistem, dan dapat dilakukan dengan mengganti *hardware*, *software* ataupun metode agar sistem lebih baik lagi.



## BAB 4 REKAYASA PERSYARATAN

Pada bab ini akan dijelaskan satu per satu secara rinci mengenai gambaran umum sistem, analisis kebutuhan fungsional, kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengumpulan data.

### 4.1 Gambaran Umum Sistem

Pada gambaran umum dari sistem ini, berisi tentang tujuan, manfaat, karakteristik pengguna, lingkup operasi, Batasan sistem, dan asumsi ketergantungan dari sistem.

#### 4.1.1 Tujuan

Tujuan dari sistem yang dibuat adalah sistem berupa sebuah konveyor dan pendorong keripik kentang yang dapat menyortir atau mengklasifikasi kentang secara otomatis. Jenis keripik yang diklasifikasi adalah keripik kentang A, AB dan Super.

#### 4.1.2 Karakteristik Pengguna

Karakteristik pengguna yang dapat menggunakan sistem yang dibuat ini adalah semua orang dengan segala umur, laki-laki ataupun perempuan, fisik tidak berpengaruh dalam penggunaan karena hanya menekan tombol *Run*.

#### 4.1.3 Lingkungan Operasi

Pada Subbab ini, difokuskan pada bagaimana sistem mengklasifikasi jenis keripik kentang. Perancangan sistem dirancang secara maksimal agar hasil klasifikasi sesuai dengan diharapkan. Persyaratan lingkungan operasi sistem yang harus dilakukan adalah sebagai berikut :

1. Webcam dipasang pada jalan masuk keripik kentang dengan menghadap ke bawah yaitu belt konveyor yang bertujuan untuk mengambil citra dari kentang untuk diolah menggunakan *Background Subtraction* yang kemudian diambil nilai  $w$  dan  $h$  dari *boundingbox* objek yang terdeteksi
2. Webcam harus terhubung ke Raspberry Pi yang bertujuan untuk proses pengolahan dari gambar yang ditangkap.
3. Stepper harus terhubung ke Arduino dengan tujuan untuk menjalankan belt dari konveyor. Kecepatan dari stepper harus stabil agar pengambilan data dan timing dari aktuator pendorong tepat dan sesuai dengan yang diharapkan.

#### 4.1.4 Batasan Perancangan dan Implementasi

Dalam pembuatan sistem penyortir keripik kentang ini mempunyai beberapa batasan, batasan dari sistem sebagai berikut :

1. Sistem ini mengklasifikasi 3 jenis keripik kentang yaitu A, AB dan Super.
2. Hasil klasifikasi dapat dilihat pada monitor komputer/laptop

3. Keripik kentang yang diklasifikasi berasal dari Pabrik Keripik Kentang Istana Batu.
4. Alat yang dibuat berupa konveyor dengan belt berwarna biru polos. Menggunakan warna lain sebenarnya bisa, asalkan tidak sewarna dengan keripik kentang.
5. Raspberry Pi dan Arduino saling terhubung untuk menjalankan actuator pendorong menggunakan komunikasi I2C.
6. Perlu catu daya yang cukup untuk menghidupkan semua komponen agar stabil. Pada sistem ini menggunakan tambahan *power supply* untuk catudaya semua komponen.

#### 4.1.5 Asumsi dan Ketergantungan

Asumsi dan ketergantungan pada sistem ini antara lain sebagai berikut :

1. Pin dari setiap komponen dihubungkan ke Arduino dan raspberry Pi sesuai fungsi yang dianjurkan dalam *datasheet*.
2. Webcam harus dipasang tepat ke bawah secara lurus, jika tidak maka akan mempengaruhi hasil dari klasifikasi.

#### 4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mendapatkan semua kebutuhan-kebutuhan yang dibutuhkan untuk mendukung perancangan sistem ini. Dalam analisis kebutuhan sistem dibagi menjadi dua, yakni kebutuhan fungsional dan kebutuhan non fungsional, dimana kebutuhan non fungsional terdiri kebutuhan perangkat keras dan kebutuhan perangkat lunak.

##### 4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional yang dilakukan untuk sistem meliputi :

1. Webcam  
Kamera webcam digunakan untuk mengambil citra dari keripik kentang kemudian mengolahnya pada opencv di raspberry menggunakan *Background Subtraction* sehingga akan mendapatkan suatu fitur yang akan dijadikan acuan untuk pengklasifikasian.
2. Raspberry Pi 3 dapat mengirim data ke arduino untuk menggerakkan servo  
Raspberry Pi 3 menerima dan memproses data yang didapatkan kemudian mengirim data untuk menjalankan actuator berupa servo ke arduino yang dihubungkan ke pin GPIO Raspberry Pi dengan menggunakan *interface* I2C.
3. Data dari citra keripik kentang diolah dan diambil beberapa fitur untuk basis pengklasifikasian, kemudian hasilnya akan menjadi kelas-kelas jenis kentang setelah dikasifikasi menggunakan metode *Naive Bayes*.

Kebutuhan fungsional ini berguna untuk pengklasifikasian berdasarkan parameter uji yang didapatkan dari pengolahan citra kentang dan data latih yang telah disimpan pada program *Naive Bayes*.

#### 4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional dari sistem pemilah keripik kentang terdiri dari kebutuhan perangkat keras dan perangkat lunak.

##### 4.2.2.1 Kebutuhan Perangkat Keras

Untuk mendukung implementasi sistem ini, maka diperlukan perangkat keras yang dijelaskan berikut ini:

1. Webcam  
Alat ini digunakan untuk mengambil video
2. Raspberry Pi 3  
Raspberry Pi 3 digunakan untuk pusat pengolahan dan pemrosesan input dan output data dalam sistem.
3. Arduino uno  
Arduino uno berfungsi sebagai penggerak motor stepper dan penggerak servo untuk mendorong keripik kentang
4. Stepper motor  
Stepper digunakan untuk penggerak belt dari konveyor
5. LED  
Lampu LED digunakan untuk membantu pencahayaan dari pengambilan citra dari keripik kentang yang diletakkan pada sebelah kamera
6. Motor Servo  
Motor servo digunakan untuk penggerak aktuator yang akan mendorong kentang sesuai dengan kelasnya.
7. Laptop  
Laptop disini berfungsi sebagai media untuk proses pemrograman Raspberry Pi dan Arduino
8. Monitor  
Laptop disini berfungsi sebagai media untuk menampilkan tampilan dari Raspberry Pi

##### 4.2.2.2 Kebutuhan Perangkat Lunak

Dalam implementasi sistem ini, diperlukan perangkat lunak sebagai pendukungnya yang dijelaskan berikut ini:

1. VNC server  
Perangkat lunak yang digunakan untuk menampilkan GUI dari Raspberry
2. MobaXTerm

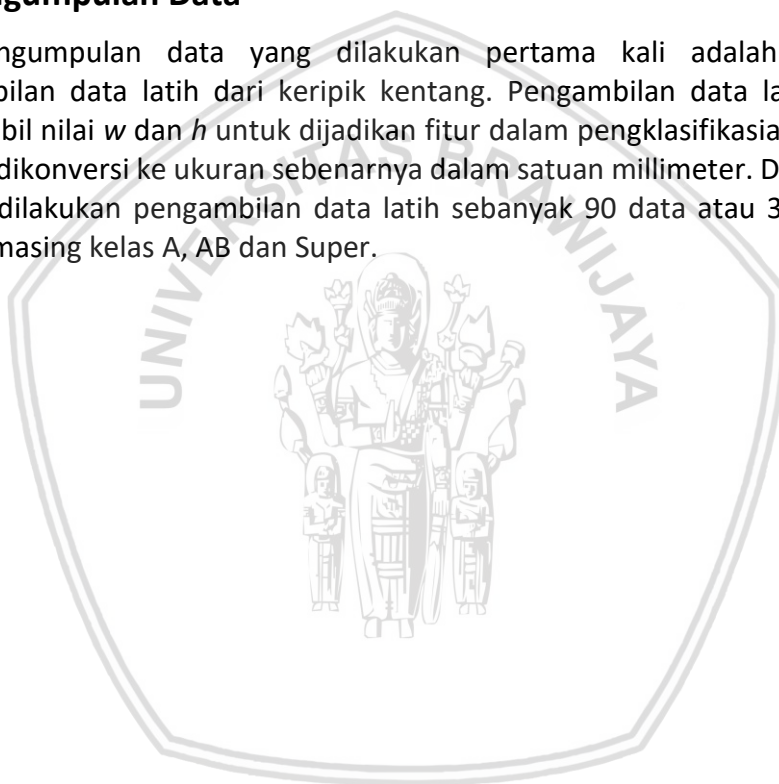
Aplikasi yang digunakan untuk meremote *command line* dari Raspberry Pi dengan protokol SSH.

### 3. *Library*

Beberapa *library* yang digunakan untuk mendukung program dalam Raspberry Pi yakni, "RPi.GPIO", library yang berfungsi agar GPIO dari Raspberry Pi dapat digunakan di dalam program. *Library "time"*, berfungsi dalam hal perwaktuan program, contohnya memberikan delay waktu pada program. *Library "math"*, ini berfungsi untuk melakukan perhitungan-perhitungan matematika yang rumit.

## 4.3 Pengumpulan Data

Pengumpulan data yang dilakukan pertama kali adalah melakukan pengambilan data latih dari keripik kentang. Pengambilan data latih tersebut mengambil nilai  $w$  dan  $h$  untuk dijadikan fitur dalam pengklasifikasian. Kemudian  $w$  dan  $h$  dikonversi ke ukuran sebenarnya dalam satuan millimeter. Dari fitur yang didapat dilakukan pengambilan data latih sebanyak 90 data atau 30 data pada masing-masing kelas A, AB dan Super.





## BAB 5 PERANCANGAN DAN IMPLEMENTASI

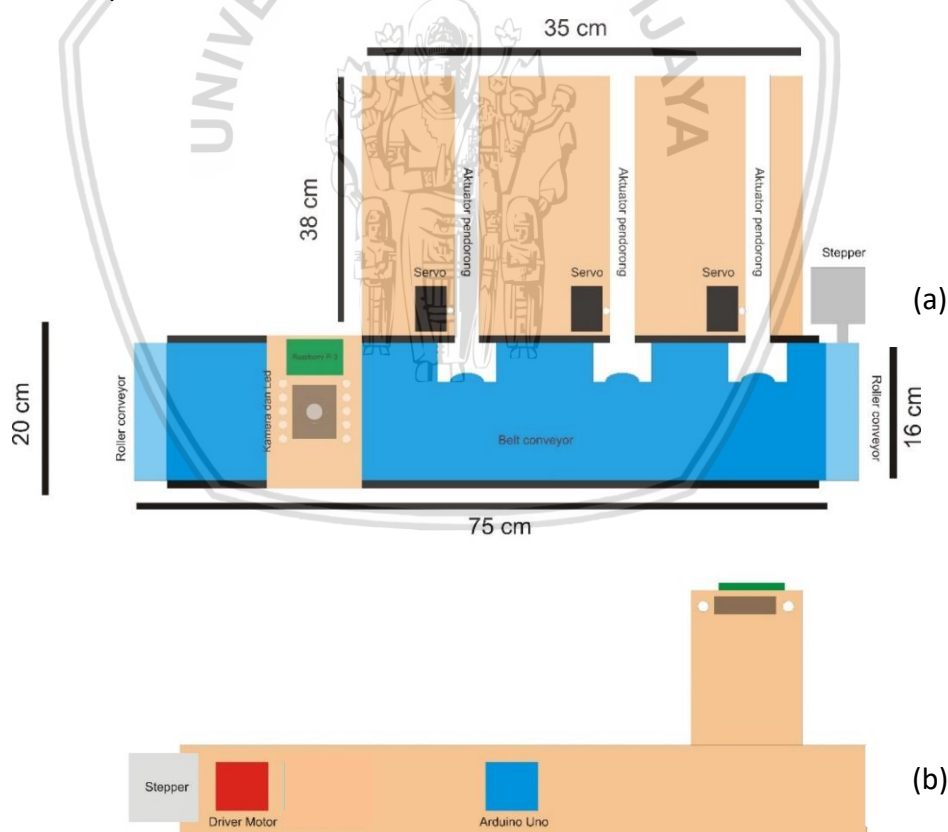
Pada bab perancangan dan implementasi menjelaskan secara rinci proses perancangan dan implementasi sistem, baik perancangan dan implementasi dari sisi perangkat keras maupun dari perangkat lunak.

### 5.1 Perancangan Sistem

Dalam perancangan sistem akan dijelaskan cara merancang sistem mulai dari perancangan *prototype*, kemudian perangkat keras dan perancangan perangkat lunak.

#### 5.1.1 Perancangan *Prototype* Alat Pemilah Keripik Kentang

Sebelum dilakukan pembuatan alat ini perlu dirancang peletakkan tiap komponen elektronik dan ukuran dari bagian pembangun alat ini. Pembuatan *desain* dari sistem dibuat menggunakan aplikasi CorelDraw untuk membuat bentuk gambar keseluruhan dari alat yang berupa konveyor. Bentuk *prototype* alat dapat dilihat pada Gambar 5.1 berikut.



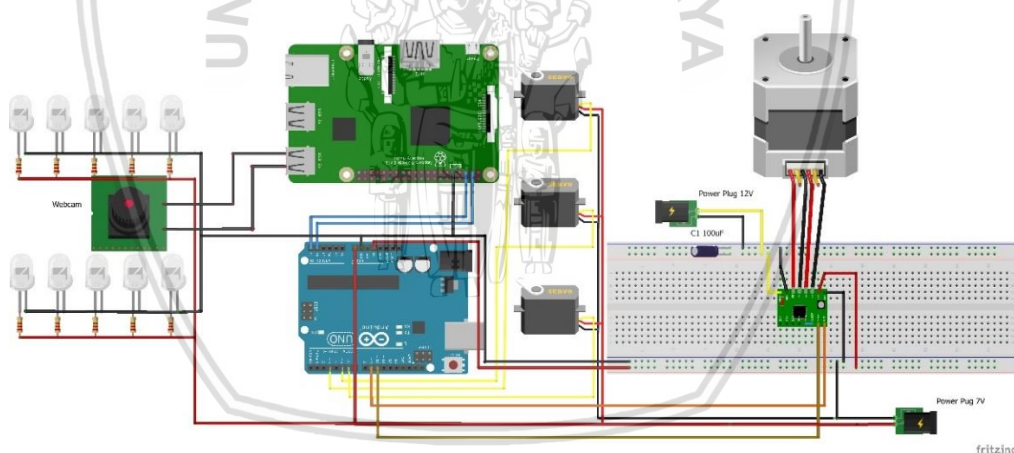
**Gambar 5.1 Desain *Prototype* Alat Pemilah Kentang Otomatis**

- (a) Gambar konveyor tampak atas
- (b) Gambar konveyor tampak samping

Posisi webcam berada dibawah menghadap belt konveyor dan dibantu led untuk pencahayaan pengambilan citra dari keripik. Keripik akan dijalankan konveyor menuju ke webcam kemudian akan diklasifikasi oleh aktuator yang digerakkan servo menurut jenisnya. Konveyor sendiri terdiri dari belt dan roller yang digerakkan oleh motor Stepper.

### 5.1.2 Perancangan Perangkat Keras

Setelah dilakukan analisis kebutuhan perangkat keras maka akan digunakan sebagai acuan untuk perancangan perangkat keras sesuai dengan spesifikasi alat agar sistem berjalan sesuai dengan apa yang diharapkan. Berdasarkan pada Gambar 3.2 blok diagram sistem, maka pada tahap perancangan perangkat keras ini menjelaskan secara rinci koneksi antar komponen atau perangkat pada masing-masing pin yang digunakan, dalam hal ini webcam merupakan input dihubungkan dengan Raspberry Pi sebagai pengolah data sehingga nanti hasilnya untuk menentukan jenis kentang yang akan disortir secara otomatis oleh servo yang dihubungkan ke Arduino Uno yang juga terhubung ke Raspberry menggunakan komunikasi i2c, pada konveyor sendiri digerakkan oleh stepper yang terhubung ke arduino uno untuk mengatur speed berputarnya roller yang menggerakkan Belt. Skematik perancangan perangkat keras Sistem pemilah kentang ini ditunjukkan pada Gambar 5.2.



Gambar 5.2 Skematik Rangkaian Sistem

Pada rangkaian skematik diatas terlihat bahwa stepper berfungsi sebagai penggerak belt pada konveyor. Stepper Nema 17 dihubungkan ke A4988 Stepper Motor Driver kemudian driver dihubungkan ke Arduino Uno. Selanjutnya rangkaian terkait koneksi antar pin akan dijelaskan pada Tabel berikut.

Tabel 5.1 Keterangan Koneksi Driver Motor dengan Power Supply

Pin Driver Motor	Power Supply
VMod	12 V
GND	GND

**Tabel 5.2 Keterangan Koneksi Arduino dengan Power Supply**

Arduino	Power Supply
Vin	7 V
GND	GND

**Tabel 5.3 Keterangan Koneksi Stepper dengan Driver Motor**

Pin Stepper	Pin Driver Motor
2A	2A
2B	2B
1A	1A
1B	1B

**Tabel 5.4 Keterangan Koneksi Driver Motor dengan Arduino Uno**

Pin Driver Motor	Pin Arduino Uno
VDD	5 V
GND	GND
STEP	D10
DIR	D9

Berdasarkan pada Tabel 5.1 terlihat pin dari VMod dan GND dari driver tersambung ke catudaya 12 V dari power supply yang berguna untuk catu daya dari stepper. Pada Arduino Uno sendiri Vin terhubung ke catu daya 7 V untuk menyalakan arduino yang digambarkan pada Tabel 5.2. Kemudian pada Tabel 5.4 terlihat pin dari stepper yang terdiri dari 4 pin terhubung ke pin dari Driver Motor, masing pada pin 2A, 2B, 1A, 1B. Kemudian pada Tabel 5.3 terlihat bahwa pin VDD dari driver Motor dihubungkan ke VCC 5 V dan GND dari arduino, kemudian pada pin STEP dihubungkan dengan pin digital dari arduino yaitu pin D10 dan pin DIR dihubungkan ke pin D9 kedua berfungsi sebagai mengatur arah putaran dan kecepatan putar dari stepper.

**Tabel 5.5 Keterangan Koneksi Raspberry dan Arduino Uno**

Pin GPIO Raspberry	Pin Arduino Uno
GPIO 3 (SDA)	A4 (SDA)
GPIO 5 (SCL)	A5 (SCL)

Pada Tabel 5.6 terlihat bahwa dari pin GPIO 3 yang merupakan interface I2C SDA terhubung pada pin analog A4 pada arduino yang merupakan pin SDA dari arduino dan pin GPIO 5 yang merupakan I2C SCL dari Raspberry dihubungkan ke pin A5 dari Arduino yang merupakan pin SCL dari Arduino, yang memungkinkan keduanya bisa saling mengirim dan menerima data, dalam sistem ini nantinya Raspberry mengirim perintah untuk menjalankan servo sesuai dengan klasifikasi keripik kentang.

**Tabel 5.6 Keterangan koneksi pin servo dengan power supply**

Pin Servo	Power Supply
VCC servo 1	7 V
GND servo 2	GND
VCC servo 2	7 V
GND servo 2	GND
VCC servo 3	7 V
GND servo 3	GND

**Tabel 5.7 Keterangan koneksi pin Arduino Uno dengan Servo**

Pin Servo	Pin Arduino Uno
Pin signal servo 1	D3
Pin signal servo 2	D5
Pin Signal servo 3	D6

Pada Tabel 5.7 menunjukkan koneksi antara pin servo dan power supply, VCC dan GND dari masing-masing servo disambungkan ke sumber tegangan 7 V dari power supply secara parallel. Kemudian pada Tabel 5.8 menunjukkan pin pulse signal dari servo 1 dihubungkan ke pin digital PWM D3 arduino, lalu pin signal servo 2 dihubungkan ke pin D5 arduino dan servo 3 dihubungkan ke pin D6.

**Tabel 5.8 Keterangan koneksi Webcam dengan Raspberry Pi 3**

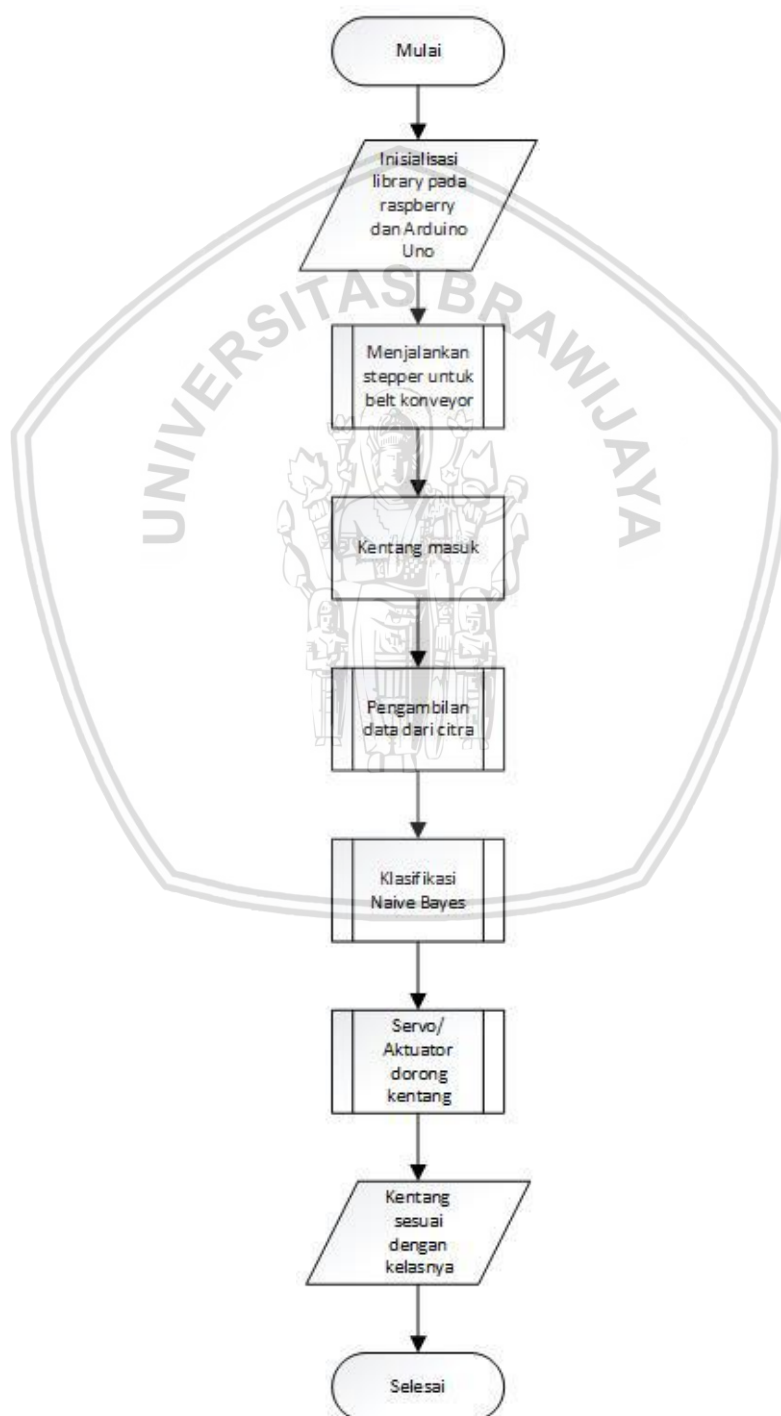
Webcam	Raspberry Pi
Webcam USB Cable	USB

Berdasarkan Tabel 5.8 terlihat bahwa kabel USB dari webcam dihubungkan ke port USB pada Raspberry Pi yang nantinya untuk mengambil citra dari keripik kentang.

### 5.1.3 Perancangan Perangkat Lunak

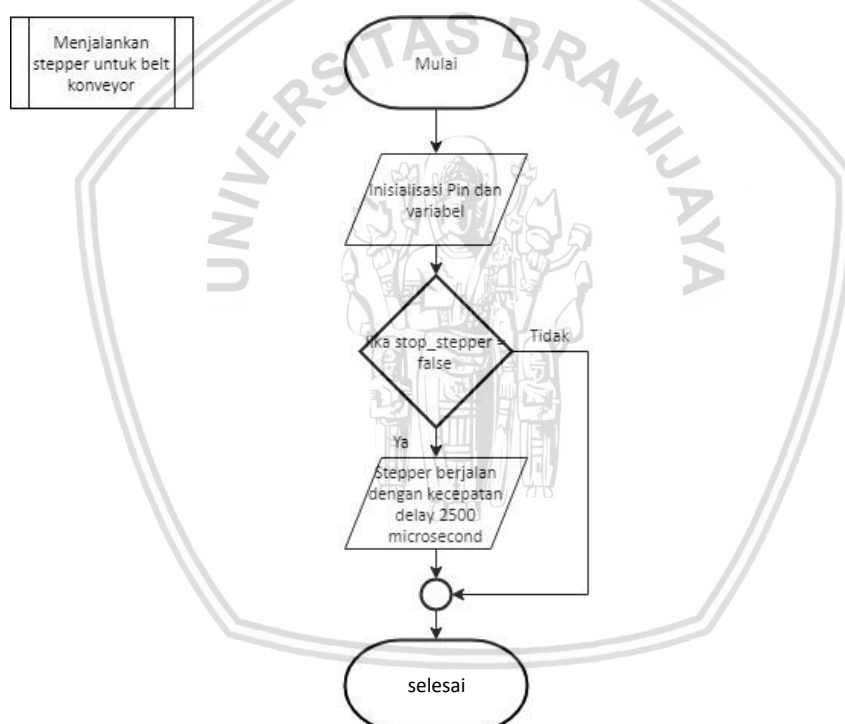
Pada sub-bab perancangan perangkat lunak terbagi menjadi beberapa pembahasan, yaitu perancangan pada pengambilan data pada mini-PC yang akan melakukan proses pengklasifikasian menggunakan metode *Naïve Bayes*.

#### 5.1.3.1 Perancangan Program Utama Sistem



**Gambar 5.3 Diagram Alir Program Utama Sistem**

Proses perancangan program utama ditunjukkan pada Gambar 5.3 diatas, hal tersebut menggambarkan gambaran sistem secara keseluruhan. Dimulai dari inialisasi pada komponen Arduino Uno dan Raspberry Pi, kemudian stepper berjalan menggerakkan belt dari konveyor, lalu kentang dimasukkan ke konveyor dan diambil gambarnya oleh webcam kemudian diolah oleh Raspberry Pi untuk mendapatkan data berupa Panjang dan lebar dalam hal ini disimpan dalam variable *wmm* dan *hmm* yang akan diolah yang didapat dari pengolahan gambar menggunakan *Background Subtraction* dan deteksi *countour* yang diberi *boundingbox*, setelah mendapat nilai data dari kemudian diklasifikasi menggunakan metode *Naïve Bayes* untuk menentukan kelas kentang, setelah mendapatkan kelasnya Raspberry mengirim perintah ke Arduino menggunakan komunikasi I2C (*Inter Integrated Circuit*) untuk mendorong kentang sesuai dengan kelasnya, keripik kentang akan terpilah sesuai dengan kelasnya.

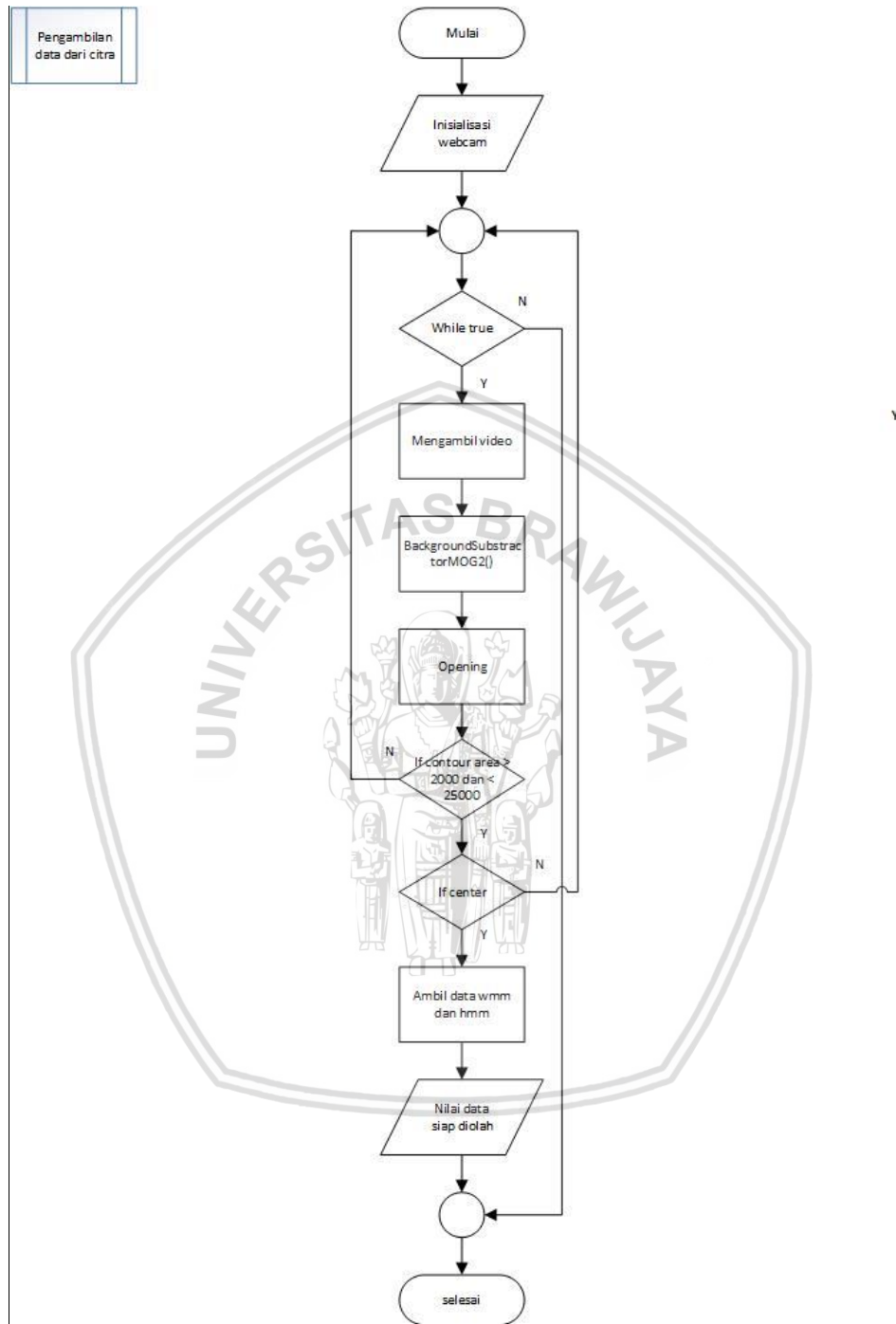


**Gambar 5.4 Diagram Alir Perancangan Program Stepper**

Pada diagram alir program stepper yang diilustrasikan pada Gambar 5.4 dimulai dengan inialisasi pin yang digunakan pada arduino dan variabel yang digunakan dalam *stepper*, lalu jika ada masukan dari Raspberry yang disimpan dalam variable number, jika number = 5 maka stepper berjalan jika number = 4 maka akan berhenti, kemudian menjalankan *stepper* dengan *delay 2500 microsecond* secara berulang sampai program dihentikan oleh *user*.



### 5.1.3.2 Perancangan Pengambilan Data dari Citra

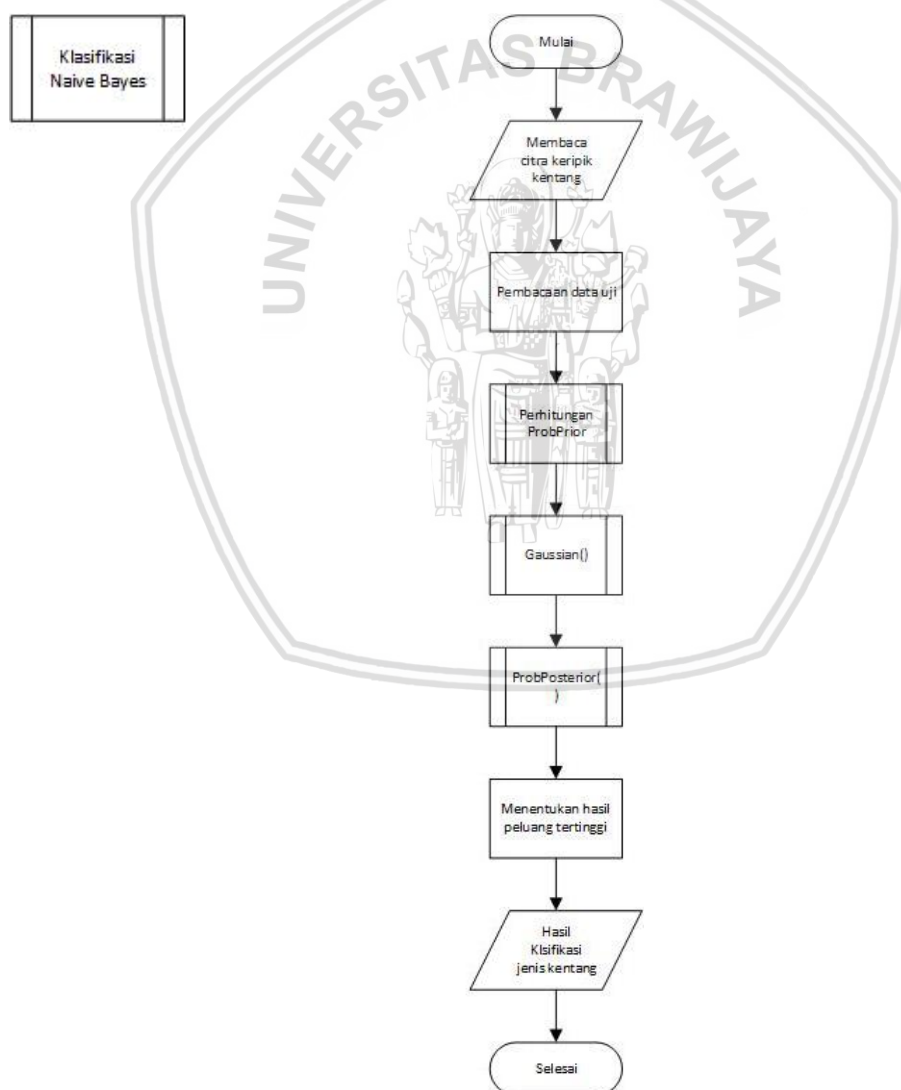


**Gambar 5.5 Diagram alir perancangan perangkat lunak pengambilan data dari citra**

Pada perancangan pengambilan data ditunjukkan pada Gambar 5.5 diatas, maksudnya adalah sistem membaca data nilai dari citra keripik kentang untuk dijadikan fitur-fitur pengklasifikasian menggunakan metode *naïve Bayes*, Raspberry akan melakukan pembacaan citra melalui webcam secara terus

menerus sampai ada *interrupt* dari *keyboard* atau menghentikan program dari aplikasi. Sistem dimulai dari inisialisasi webcam yang digunakan sebagai input, kemudian webcam akan mengambil citra secara terus menerus sampai program dihentikan, citra tersebut dimasking dengan library BackgroundSubtractorMOG2, untuk mengurangi noise ditambahkan opening dan dimasking ke *frame* dari *backgroundsubtractor*, Kemudian memberikan bounding pada area *pixel* yang mempunyai nilai area  $> 2000$  dan area  $< 25000$  nilai yang diambil adalah antara nilai tersebut nilai yang terlalu kecil ( $> 2000$ ) atau terlalu besar tidak akan dideteksi boundingbox, saat pada posisi center akan didapat nilai panjang (wmm) dan tinggi (hmm). Nilai yang didapatkan dari citra keripik kentang akan digunakan untuk perhitungan *Naïve Bayes* dalam menentukan klasifikasi jenis keripik kentang yang akan diuji.

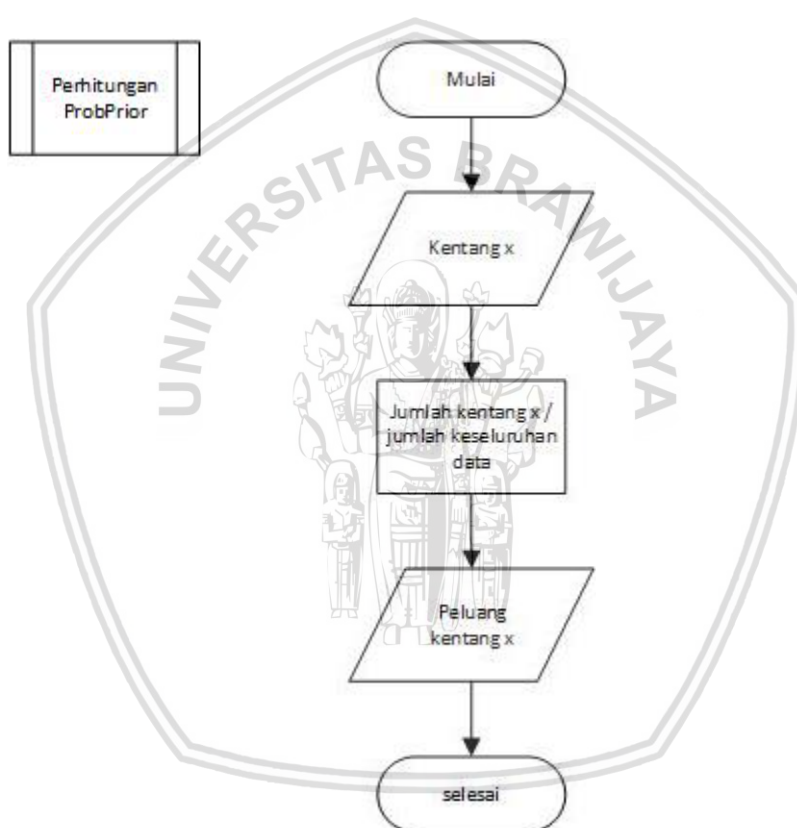
### 5.1.3.3 Perancangan Klasifikasi *Naïve Bayes*



Gambar 5.6 Diagram alir perancangan klasifikasi *Naïve Bayes*

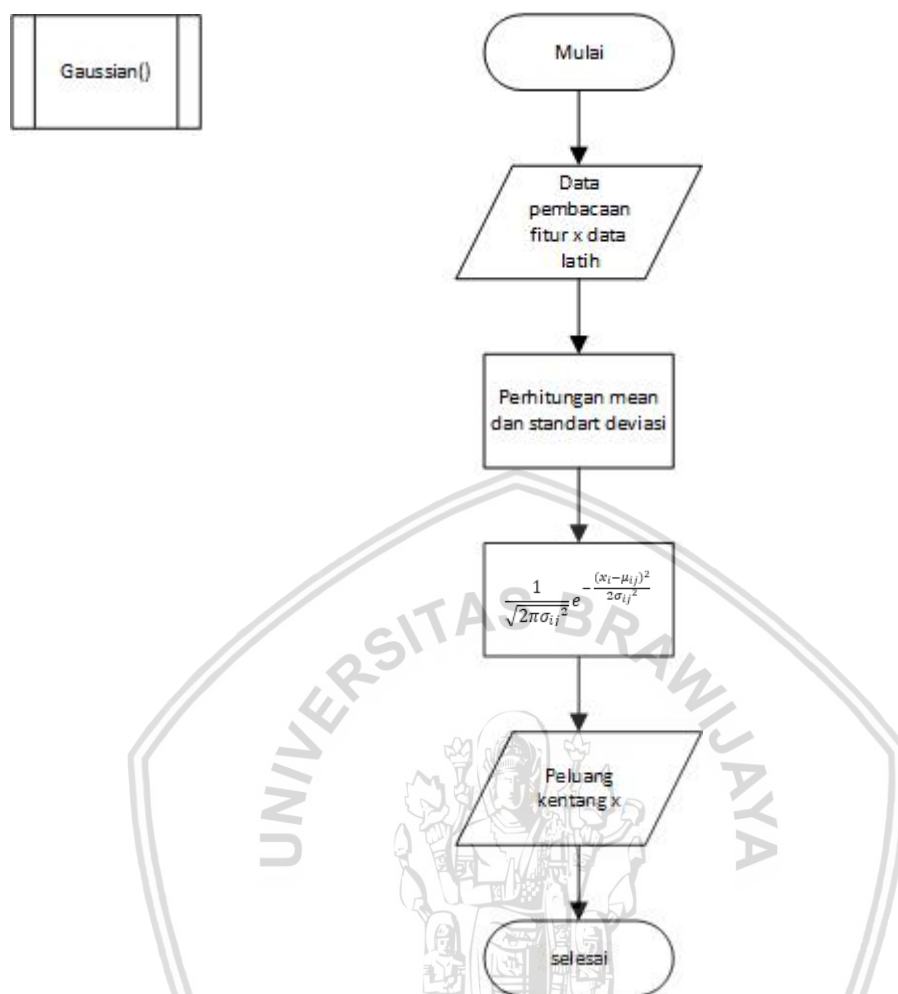
Pada proses klasifikasi *Naïve bayes* terdapat beberapa tahapan seperti pada Gambar 5.6, nilai dari pembacaan citra dari webcam akan dijadikan masukan. Kemudian hasil pembacaan citra keripik kentang yang dijadikan fitur-fitur yang akan mempengaruhi hasil klasifikasi jenis keripik kentang, selain itu hasil data latih yang diambil juga mempengaruhi hasil klasifikasi. Proses dimulai dengan menentukan nilai prior, lalu menentukan nilai hasil fungsi Gaussian() kemudian menentukan fungsi ProbPosterior(), menentukan hasil peluang tertinggi hingga didapat hasil klasifikasi jenis keripik kentang. penjelasan dari masing-masing fungsi diilustrasikan pada diagram alir pada

Gambar 5.6.



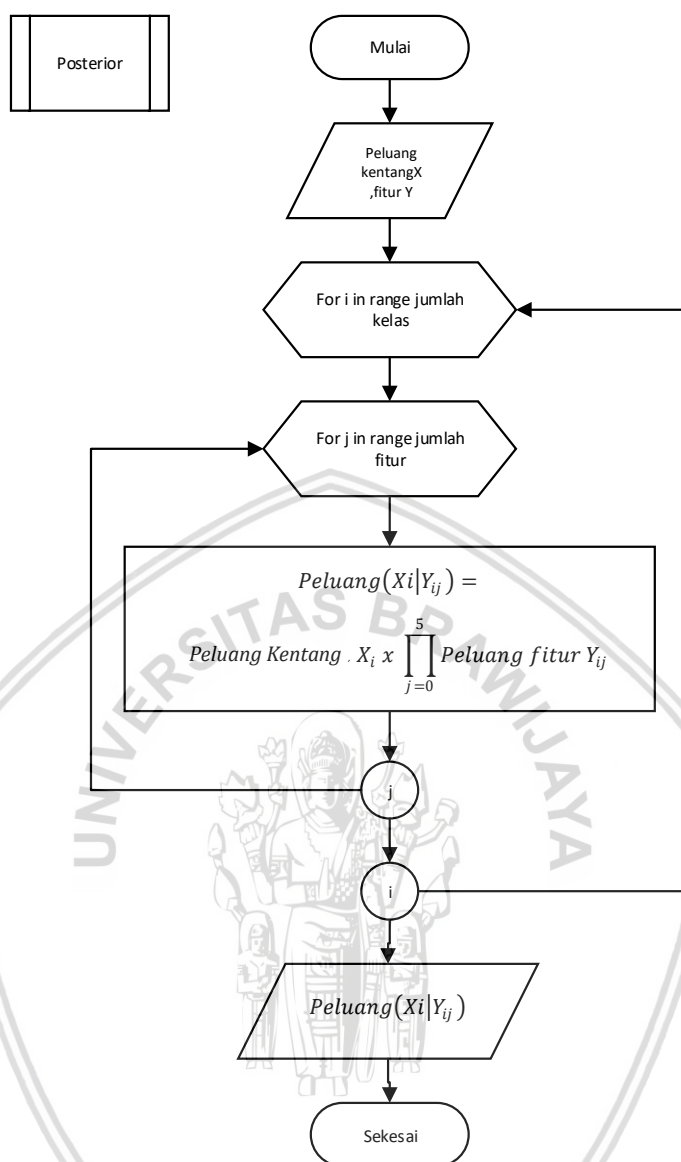
Gambar 5.7 Diagram alir ProbPrior

Hal yang dilakukan pertama kali dalam mengklasifikasi keripik kentang menggunakan metode *Naïve Bayes* adalah menghitung nilai prior dari tiap-tiap kelas keripik kentang dengan menggunakan diagram alir yang diilustrasikan pada Gambar 5.7. Nilai *prior* adalah nilai peluang terjadinya suatu kelas dibagi banyaknya data dalam suatu kelas (dalam sistem ini terdapat 3 kelas yaitu keripik kentang A, AB dan SUPER) dengan jumlah keseluruhan data. Data yang digunakan untuk perhitungan *prior* berasal dari data latih yang diambil.



**Gambar 5.8 Diagram alir fungsi Gaussian()**

Tahapan kedua yaitu menentukan nilai peluang dari setiap fitur. Pada sistem ini terdapat 2 fitur yang digunakan yaitu *wmm* yang merupakan panjang dari objek dan *hmm* yang merupakan tinggi dari objek yang diukur dalam satuan millimeter. Tetapi sebelum menentukan nilai fitur, terlebih dahulu harus menghitung nilai *mean* dan standart deviasi dari data latih menggunakan **Persamaan (2.4)** dan **Persamaan (2.5)**. Dalam sistem ini nilai *mean* dan *standart deviasi* disimpan dalam code program raspberry untuk memudahkan dalam mengakses nilai dari data latih saat dijalankan, tetapi dalam menghitung nilai *mean* dan *standart deviasi* menggunakan aplikasi Ms Excel. Kemudian menghitung nilai *Gaussian* dengan menggunakan **Persamaan (2.3)** seperti pada Gambar 5.8 dimana nilai *x* adalah nilai fitur hasil pembacaan pembacaan citra.



**Gambar 5.9 Diagram alir ProbPosterior()**

Tahapan berikutnya adalah menentukan peluang *posterior* yang diilustrasikan pada Gambar 5.9, yaitu untuk menentukan besarnya peluang dari masing-masing kelas ketika adanya inputan dari fitur yang masuk. Sistem ini menentukan besarnya peluang masing-masing jenis keripik kentang ketika adanya pembacaan nilai fitur *wmm* dan *hmm*. Prosesnya adalah melakukan perkalian antara hasil dari *Gaussian* dan *Posterior*.

Tahap terakhir klasifikasi *Naive Bayes* yaitu dengan menentukan nilai *posterior* yang tertinggi dengan cara membandingkan nilai masing-masing kelas antara peluang *posterior*, nilai yang tertinggi adalah nilai *posterior* dari klasifikasi keripik kentang yang dibaca oleh sistem.

Perhitungan manual klasifikasi terhadap keripik kentang dengan data uji yang mempunyai fitur  $wmm = 22$  dan  $hmm = 32$  berdasarkan data latih pada lampiran adalah sebagai berikut :

1. Menghitung peluang prior dari masing-masing jenis keripik kentng

$$P_{Kentang A} = \frac{Jumlah\ data\ kentang\ A}{Jumlah\ seluruh\ data} = \frac{30}{90} = 0.333$$

$$P_{Kentang AB} = \frac{Jumlah\ data\ kentang\ AB}{Jumlah\ seluruh\ data} = \frac{30}{90} = 0.333$$

$$P_{Kentang Super} = \frac{Jumlah\ data\ kentang\ Super}{Jumlah\ seluruh\ data} = \frac{30}{90} = 0.333$$

2. Menghitung nilai mean dan standart deviasi masing-masing jenis keripik kentang dengan data Wmm dan Hmm yang terdapat pada lampiran

$$\begin{aligned}\mu_{wmm(A)} &= \frac{\sum_{i=1}^n wmm_i}{Jumlah\ data\ KentangA} \\ &= \frac{26.13917344 + 28.25856588 + 30.37795832 + \dots}{30} \\ &= 26,95160721\end{aligned}$$

$$\begin{aligned}\mu_{hmm(A)} &= \frac{\sum_{i=1}^n hmm_i}{Jumlah\ data\ KentangA} \\ &= \frac{33.91027905 + 40.26845638 + 42.03461674 + \dots}{30} \\ &= 30,778288\end{aligned}$$

Perhitungan yang sama dilakukan untuk mencari *mean* dari jenis kentang AB dan Kentang Super. Nilai selanjutnya dapat dilihat pada lampiran pada lampiran data latih. Hasil dari perhitungan mean seluruh jenis keripik kentang dapat dilihat pada Tabel 5.9.

**Tabel 5.9 Mean tiap jenis keripik kentang**

	Wmm	Hmm
Kentang A	26.95160721	30.778288
Kentang AB	37.23066054	40.11538914
Kentang Super	49.11103262	57.2118215



## Standart Deviasi

$$\sigma_{wmm(A)} = \sqrt{\frac{\sum_{i=1}^n (wmm_i - \mu_{wmm(A)})^2}{\text{Jumlah data kentang A} - 1}}$$

$$= \sqrt{\frac{(26.13917344 - 26.95160721)^2 + (28.25856588 - 26.95160721)^2 + (30.37795832 - 26.95160721)^2 + \dots}{30 - 1}}$$

$$= 5.15996$$

$$\sigma_{hmm(A)} = \sqrt{\frac{\sum_{i=1}^n (hmm_i - \mu_{hmm(A)})^2}{\text{Jumlah data kentang A} - 1}}$$

$$= \sqrt{\frac{(33.91027905 - 30.778288)^2 + (40.26845638 - 30.778288)^2 + (42.03461674 - 30.778288)^2 + \dots}{30 - 1}}$$

$$= 5.142077$$

Perhitungan yang sama dilakukan untuk mencari standar deviasi dari jenis kentang AB dan kentang Super. Nilai selanjutnya dapat dilihat pada lampiran pada lampiran data latih. Hasil perhitungan standar deviasi keseluruhan jenis kentang ditunjukkan pada Tabel 5.7.

**Tabel 5.10 Tabel standart deviasi tiap jenis keripik kentang**

	$\sigma$ Wmm	$\sigma$ Hmm
Kentang A	5.159966	5.142077
Kentang AB	6.50442	6.391709
Kentang Super	16.29323	13.90258

3. Menghitung nilai gaussian dari masing-masing fitur yang diujikan ( $wmm=22$ ,  $hmm=32$ )

$$P_{(Wmm=22|KentangA)} = \frac{1}{\sqrt{2\pi\sigma_{Wmm(KentangA)}^2}} e^{-\frac{(X-\mu_{Wmm(KentangA)})^2}{2\sigma_{Wmm(KentangA)}^2}}$$

$$= \frac{1}{\sqrt{2 \times 3,14 \times 5.159966^2}} e^{-\frac{(22-26.95160721)^2}{2 \times 5.159966^2}} = 0,048799$$

$$P_{(Wmm=22|KentangAB)} = \frac{1}{\sqrt{2\pi\sigma_{Wmm(KentangAB)}^2}} e^{-\frac{(X-\mu_{Wmm(KentangAB)})^2}{2\sigma_{Wmm(KentangAB)}^2}}$$

$$= \frac{1}{\sqrt{2 \times 3,14 \times 6.50442^2}} e^{-\frac{(22-37.23066054)^2}{2 \times 6.50442^2}} = 0.003955$$

$$P_{(Wmm=22|KentangSuper)} = \frac{1}{\sqrt{2\pi\sigma_{Wmm(KentangSuper)}^2}} e^{-\frac{(X-\mu_{Wmm(KentangSuper)})^2}{2\sigma_{Wmm(KentangSuper)}^2}}$$

$$= \frac{1}{\sqrt{2 \times 3,14 \times 16.29323^2}} e^{-\frac{(22-49.11103262)^2}{2 \times 16.29323^2}} = 0.006135$$

Untuk mencari nilai gaussian dari fitur Hmm dilakukan perhitungan yang sama seperti diatas. Hasil dari perhitungan nilai Gaussian keseluruhan jenis keripik kentang dituliskan pada Tabel 5.11.

**Tabel 5.11 Tabel gaussian tiap jenis keripik kentang**

	<i>P</i> Wmm	<i>P</i> Hmm
Kentang A	0.048799	0.018073
Kentang AB	0.003955	0.001125
Kentang Super	0.006135	0.001161

4. Menghitung nilai posterior dari masing-masing fitur yang diujikan (wmm = 22, hmm = 32)

$$P_{(kentangA|Wmm=22,Hmm=32)}$$

$$= P_{KentangA} \times P_{(Wmm=22|kentangA)} \times P_{(Hmm=32|kentangA)}$$

$$= 0.333 \times 0.048799 \times 0.018073 = 0.000293691$$

$$P_{(kentangAB|Wmm=22,Hmm=32)}$$

$$= P_{KentangAB} \times P_{(Wmm=22|kentangAB)} \times P_{(Hmm=32|kentangAB)}$$

$$= 0.333 \times 0.003955 \times 0.001125 = 1.48169 \times 10^{-6}$$

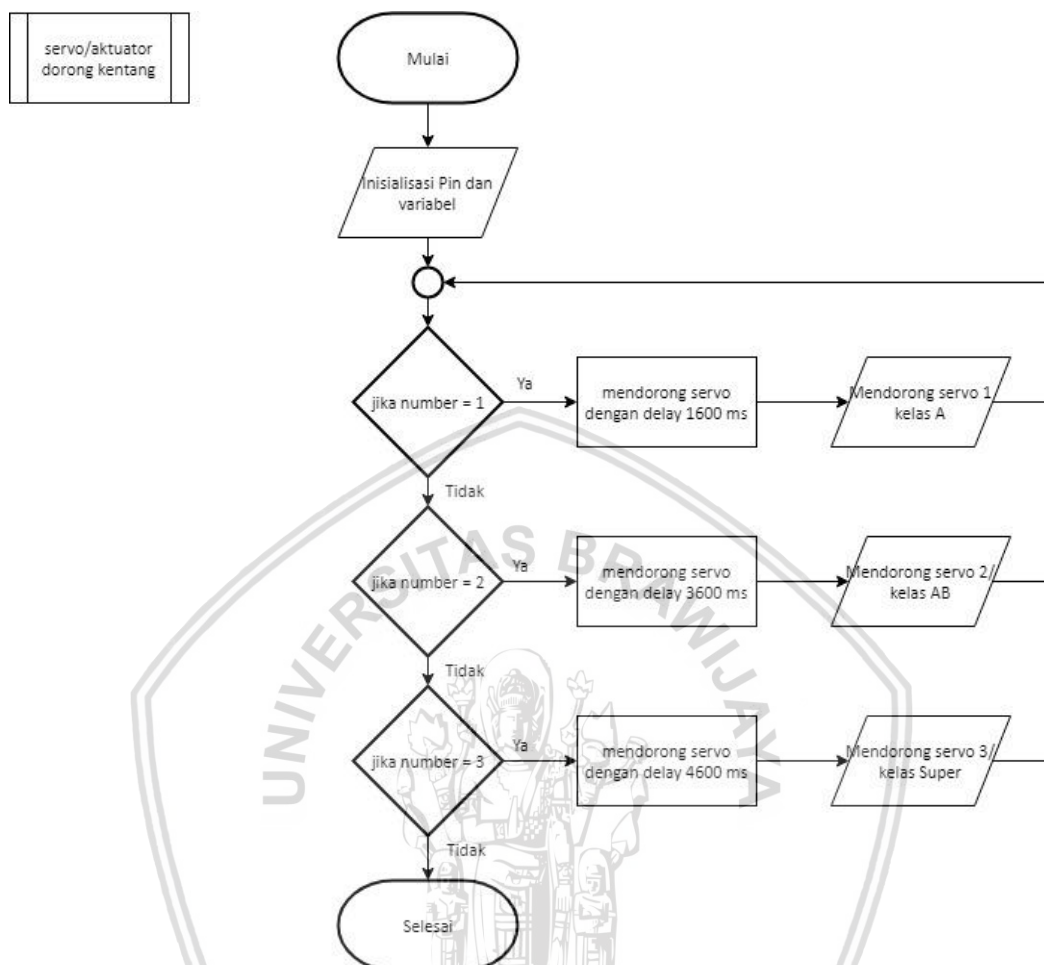
$$P_{(kentangS|Wmm=22,Hmm=32)}$$

$$= P_{KentangS} \times P_{(Wmm=22|kentangS)} \times P_{(Hmm=32|kentangS)}$$

$$= 0.333 \times 0.006135 \times 0.001161 = 2.37245 \times 10^{-6}$$

Berdasarkan pada perhitungan peluang posterior diatas, yang memiliki nilai peluang posterior paling tinggi adalah jenis kentang A. Sehingga dapat disimpulkan bahwa fitur yang bernilai wmm=22, hmm=32, termasuk jenis kentang A.

### 5.1.3.4 Perancangan Servo Pemilah Kentang



Gambar 5.10 Diagram alir perancangan servo

Selanjutnya adalah merancang bagaimana servo pemilah sesuai jenis kentang, yang dapat dilihat pada Gambar 5.10 Diagram alir perancangan servo. Pada tahap awal adalah inisialisasi servo 1, 2 dan 3. Kemudian servo akan mendorong/memilah kentang sesuai dengan alamat berupa sebuah angka yang disimpan dalam variable *number* yang dikirim oleh Raspberry Pi ke Arduino.

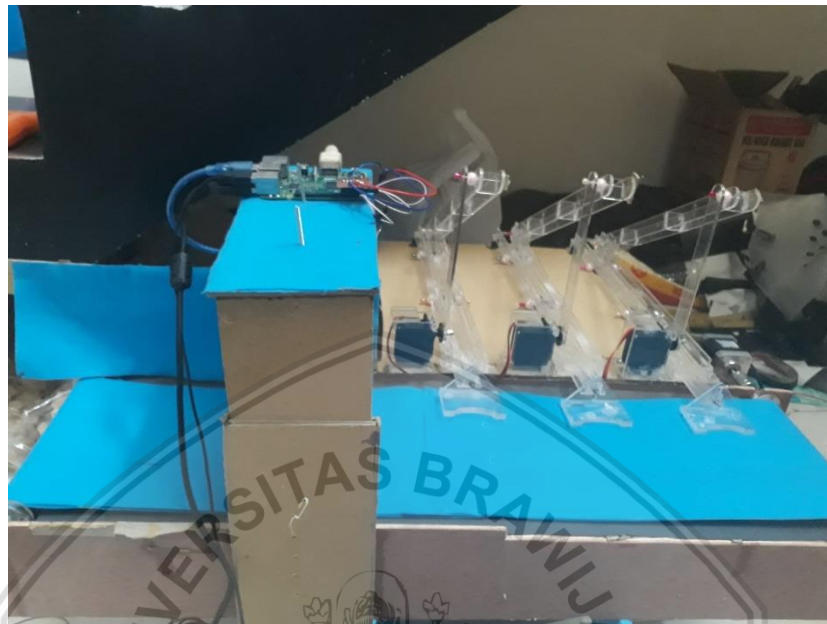
## 5.2 Implementasi Sistem

Implementasi sistem merupakan tahapan untuk merealisasikan pembuatan sistem berdasarkan semua perancangan yang telah dibuat sebelumnya. Pada subbab ini menjelaskan satu per satu secara rinci terkait implementasi prototype, implementasi perangkat keras serta implementasi perangkat lunak.

### 5.2.1 Implementasi Prototype Alat Pemilah Kentang

Dalam mengimplementasikan prototype alat pemilah ini mengacu pada perancangan di subbab 5.1.1 yakni menggunakan sebuah konveyor. Bahan yang digunakan untuk membuat konveyor adalah kayu triplek, kemudian karton, dan

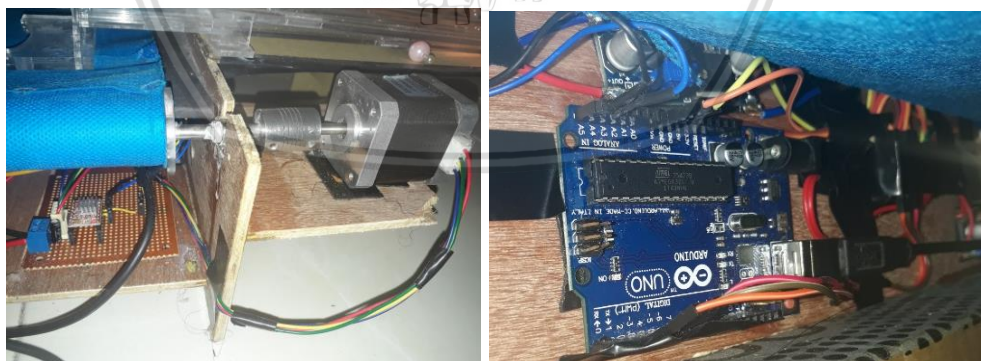
lengan pendorong terbuat dari akrilik dengan ketebalan sebesar 2 mm. Hasil implementasi prototype beserta peletakan komponen elektronik yang digunakan ditunjukkan pada Gambar 5.11.



**Gambar 5.11 Implementasi Prototype Alat Pemilah Kentang**

### 5.2.2 Implementasi Perangkat Keras

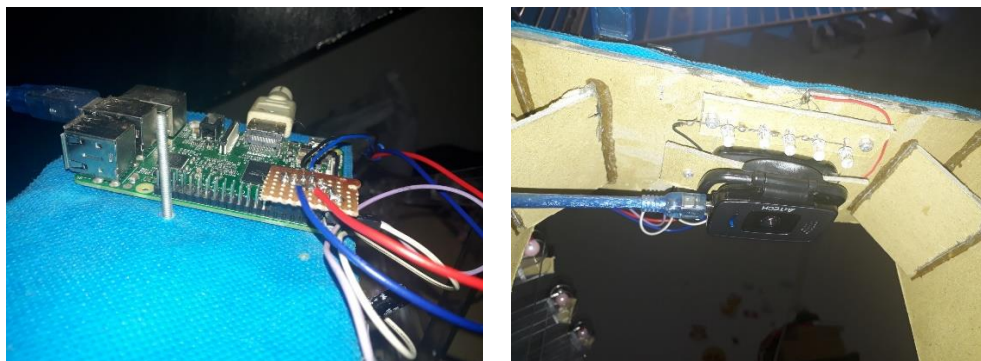
Pada tahap ini menjelaskan implementasi perangkat keras yang mencakup semua komponen atau alat yang membangun sistem ini. Keseluruhan komponen dirangkai menjadi satu didalam konveyor yang dihubungkan berdasarkan perancangan yang dijelaskan pada subbab 5.1.2.



**Gambar 5.12 Implementasi rangkaian stepper dan driver dengan Arduino Uno**

Pada Gambar 5.12 menunjukkan hasil implementasi stepper motor beserta driver dengan arduino uno dimana komponen ini adalah komponen penggerak stepper yang diletakkan pada bagian depan bawah dari konveyor dihubungkan menggunakan kabel untuk catu daya dan jumper untuk pin-pinnya.





**Gambar 5.13 Implementasi rangkaian Raspberry Pi dan Webcam serta LED**

Pada Gambar 5.13 menunjukkan hasil implementasi webcam LED dan Raspberry, led dihubungkan ke power supply kemudian webcam dihubungkan ke port USB pada Raspberry Pi menggunakan kabel USB.



**Gambar 5.14 Implementasi rangkaian Raspberry Pi dan arduino Uno**

Berdasarkan perancangan perangkat keras Raspberry Pi dan Arduino Uno dihubungkan menggunakan komunikasi I2C pada pin SDA dan SCL masing-masing alat. Hasil implementasi ditunjukkan pada Gambar 5.14.



**Gambar 5.15 Implementasi rangkaian Arduino dengan servo**

Pada perancangan perangkat keras pulse dari servo dihubungkan ke pin pwm yaitu pin 3, 5 dan 6 pada arduino, kemudian VCC dan GND dari servo dihubungkan ke 7 volt power supply, yang Ditunjukkan pada Gambar 5.15.

### 5.2.3 Implementasi Perangkat Lunak

Pada Implementasi perangkat lunak akan dijelaskan proses realisasi program untuk Sistem Pemilah Kentang ini berdasarkan perancangan yang telah dibuat sebelumnya pada subbab 5.1.3, dalam melakukan implementasi perangkat lunak ini proses pengkodean dilakukan pada Opencv Python untuk melakukan image processing dan klasifikasi, kemudian arduino IDE untuk program penggerak stepper pada konveyor dan penggerak servo yang mendorong kentang sesuai kelasnya. Pada Tabel 5.12 ditunjukkan pengimplementasian library pada sistem ini, diantaranya adalah pada baris ke-1 *numpy* as *np* adalah modul yang menyediakan atau mengenali objek matematika yang digunakan dalam melakukan perhitungan, objek utama yang disediakan NumPy adalah array yang dapat berperan sebagai matrix. Pada baris ke-2 *Library* “*cv2*” untuk memanggil library *cv2* yaitu OpenCV, pada baris ke-3 *library* “*from smbus import SMBus*” adalah *library* yang memungkinkan alat saling berkomunikasi menggunakan interface I2C, kemudian baris ke-4 *library* “*math*” untuk memudahkan perhitungan matematika yang kompleks.

**Tabel 5.12 Kode pemrograman inisialisasi library Sistem Pemilah Kentang pada Raspberry**

Baris	Kode Program
1	<code>import numpy as np</code>
2	<code>import cv2</code>
3	<code>from smbus import SMBus</code>
4	<code>import math</code>

Setelah inisialisasi *library* dilakukan dilakukan pemrograman untuk capture video dan mengeset resolusi pada kamera yaitu 480x480 *pixel*. Kode program dapat dilihat pada Tabel 5.13.

**Tabel 5.13 Pengambilan gambar video**

Baris	Kode Program
1	<code>#capture video</code>
2	<code>cap = cv2.VideoCapture(0)</code>
3	<code>cap.set(3, 480.)</code>
4	<code>cap.set(4, 480.)</code>
5	
6	<code>while(True):</code>
7	<code>    ret, frame = cap.read()</code>
8	
9	<code>    if ret == False:</code>
10	<code>        break</code>
11	
12	<code>k = cv2.waitKey(5) &amp; 0xff</code>
13	<code>    if k == 27:</code>
14	<code>        break</code>
15	<code>cap.release()</code>
16	<code>cv2.destroyAllWindows()</code>



Pada Tabel 5.13 dilakukan pengambilan gambar dari kamera webcam dengan resolusi 480 x 480 *pixel*, kemudian program akan mengambil video secara berulang sampai ada interrupt dari keyboard dengan karakter 'q' atau diberhentikan langsung dari aplikasi oleh *user*. Pengambilan gambar disimpan dalam variabel frame yang nantinya akan diolah.

**Tabel 5.14 Kode pemrograman inisialisasi library Sistem Pemilah Kentang pada Arduino Uno**

Baris	Kode Program
1	#include <Wire.h>
2	#include <Servo.h>
3	#define SLAVE_ADDR 0x24

Pada Tabel 5.14 merupakan pengimplementasian library pada Arduino Uno, *library* "Wire.h" adalah library yang didedikasikan untuk menangani protocol serial sinkron secara I2C, *library* "Servo.h" untuk mempermudah memprogram objek servo, mendefinisikan "SLAVE\_ADDR 0x24" untuk membuat I2C slave beserta alamatnya yang berfungsi untuk mengirim dan menerima data.

### 5.2.3.1 Implementasi Kode Program Stepper Motor

Pada implementasi kode program stepper mengacu pada perancangan Gambar 5.1 Diagram Alir Perancangan Program Stepper yang dapat dilihat Tabel 5.15. yang dilakukan pertama kali adalah mendeklarasikan variabel "STEPPER\_PULSE\_HIGH, STEPPER\_PULSE\_LOW" dan "stop\_stepper = true; number=0", kemudian menginisialisasi pin 9 dan 10 sebagai OUTPUT. Pin 9 nantinya dijadikan *direction* apakah stepper akan berjalan maju atau mundur, dalam sistem ini stepper diprogram *HIGH* agar output yang dihasilkan stepper bergerak kearah depan atau searah jarum jam untuk menarik *belt* dan pin 10 adalah step dari stepper. Pada baris ke-11 sampai baris ke-29 adalah kode program untuk menjalankan stepper secara berulang dengan delay 2500 *microsecond* sebagai pengatur kecepatan stepper. Stepper bergerak jika mendapat inputan dari raspberry berupa angka yang disimpan dalam variabel number, yang dituliskan pada baris ke-20 sampai baris ke-33, jika number = 4 maka stepper berhenti, jika number = 5 stepper berjalan.

**Tabel 5.15 Kode program stepper motor**

Baris	Kode Program
1	unsigned long STEPPER_PULSE_HIGH, STEPPER_PULSE_LOW;
2	stop_stepper = true; number=0;
3	void setup() {
4	// put your setup code here, to run once:
5	pinMode(9, OUTPUT); //dir
6	pinMode(10, OUTPUT); //step
7	
8	digitalWrite(9, HIGH); // arah maju

9	}
10	void loop() {
11	if (micros() >= STEPPER_PULSE_HIGH) {
12	digitalWrite(10, HIGH);
13	STEPPER_PULSE_LOW = micros() + 2500;
14	STEPPER_PULSE_HIGH = STEPPER_PULSE_LOW + 2500;
15	}
16	if (micros() >= STEPPER_PULSE_LOW) {
17	digitalWrite(10, LOW);
18	}
19	}
20	void receiveData(int byteCount) {
21	while(Wire.available()) {
22	number = Wire.read();
23	.
24	.
25	.
26	} else if (number == 4) {
27	stop_stepper = true;
28	} else if (number == 5) {
29	stop_stepper = false;
30	}
31	number = 0;
32	}
33	}

### 5.2.3.2 Implementasi Kode Program Pengambilan Data dari Citra Kentang

Pada Implementasi pengambilan gambar adalah program untuk pengolahan gambar yang menggunakan BackgroundSubtractorMOG2(), kemudian opening untuk menghilangkan noise, lalu program akan membaca contour dari area yang mempunyai *range pixel* lebih dari 2000 dan kurang dari 25000 kemudian akan diberikan *bounding box* objek yang terdeteksi. Setelah terdeteksi dilakukan *centroid* untuk mengambil titik tengah dari objek dalam *bounding rectangle* objek yang terdeteksi, pada saat objek pada posisi tengah data akan diambil. Kode program dituliskan pada Tabel 5.16.

**Tabel 5.16 Proses pengambilan data pada citra keripik kentang**

Baris	Kode Program
1	cap = cv2.VideoCapture(0)
2	cap.set(3, 480.)
3	cap.set(4, 480.)
4	# inisialisasi fungsi BackgroundSub
5	fgbg = cv2.createBackgroundSubtractorMOG2()
6	
7	#pixel permilimeter
8	ppm = 2.831
9	
10	#perulangan pengambilan frame

```

11 while(True):
12     ret, frame = cap.read()
13
14     if ret == False:
15         break
16     #backgroundSub apply
17     fgmask = fgbg.apply(frame)
18     #morph opening
19     opening=cv2.morphologyEx(fgmask, cv2.MORPH_OPEN,
20 cv2.getStructuringElement(cv2.MORPH_RECT, (5,5)),
21 iterations = 3)
22
23     #baca contour dan boundingrect
24     _, contours, _ = cv2.findContours(opening,
25 cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
26
27     for cnt in contours:
28         area = cv2.contourArea(cnt)
29         if area > 2000 and area < 25000:
30             status = 1
31             x, y, w, h = cv2.boundingRect(cnt)
32
33             cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0),1)
34             #Inisialisasi pembacaan wmm
35             wmm = w / ppm
36             hmm = h / ppm
37             #centeroid
38             center = (int(x + (w/2)), int(y + (h/2)))
39             cv2.circle(frame, center, 5, (0, 255, 0),
40 -1)
41
42             if (center[1] > (int(height/2)-7)) and
43 (center[1] < (int(height/2)+7)):
44                 status = 2
45                 print (wmm, hmm)
46             #status garis
47             if status == 1:
48                 color = (255, 0, 0)
49             elif status == 2:
50                 color = (0, 0, 255)
51             #pemberian garis pada frame
52             cv2.line(frame, (0, int(height/2)), (int(width),
53 int(height/2)), color, 5)
54             #tampilkan hasil
55             cv2.imshow('mask', opening)
56             cv2.imshow('img', frame)
57
58             k = cv2.waitKey(5) & 0xff
59             if k == 27:
60                 break
61             cap.release()
62             cv2.destroyAllWindows()

```

Pada proses pengambilan gambar video diambil dan disimpan dalam variabel frame, kemudian frame dimasking dengan *Background Subtraction* yang disimpan dalam variabel fgmask yang diinisialisasi pada baris 5, *Background Subtractor* akan membedakan *background* dan *foreground* pada frame yang diambil, *background* dalam alat ini berupa belt berwarna biru dan *foreground*nya adalah keripik kentang. Setelah dilakukan pengolahan menggunakan *Background Subtraction* kemudian dilakukan pemrograman opening pada baris 19 untuk menghilangkan noise berupa objek-objek kecil pada belt. Setelah itu dilakukan pembacaan *contour* pada objek yang kemudian diberi *Bounding Rectangle* atau kotak pada *area* yang mempunyai *pixel* antara 2000 sampai 25000 yang ditulis pada baris 25-34. Setelah itu dilakukan *centroid* atau pendeteksian titik tengah pada objek yang terdeteksi pada baris 39-44. Untuk mendapatkan nilai dari *wmm* atau *hmm* diberikan garis pada tengah frame yang dijelaskan pada program baris 57 saat titik tengah melewati garis tersebut maka nilai *wmm* dan *hmm* akan didapatkan, garis yang tidak mendeteksi akan berwarna biru, dan jika terdeteksi akan berwarna merah yang ditulis pada baris 43. Nilai *wmm* dan *hmm* berasal dari nilai *w* dan *h* dari *bounding box* kemudian dibagi dengan nilai panjang dan tinggi dari objek asli yang telah diukur dengan alat ukur manual. Nilai *wmm* dan *hmm* nantinya akan diolah untuk diklasifikasi sesuai dengan kelasnya. Pada baris 56 dan 57 untuk menampilkan frame yang asli dan yang sudah diolah untuk deteksi objek.

Perhitungan untuk mendapatkan nilai *PPM* dituliskan dalam (**Persamaan 5.1**) untuk mengetahui panjang dan lebar kentang sesuai dengan aslinya adalah sebagai berikut:

$$\begin{aligned} PPM(\text{Pixel Per Milimeter}) &= \frac{\text{nilai } W \text{ dari bounding box}}{\text{Pembacaan alat ukur}} \\ &= \frac{126}{44,5} \\ &= 2,831 \end{aligned}$$

Kemudian untuk mendapatkan hasil ukuran kentang sebenarnya menggunakan (**Persamaan 5. 2**) sebagai berikut :

$$\begin{aligned} W_{mm} &= \frac{\text{Nilai } w \text{ dari bounding}}{PPM} \\ &= \frac{126}{2,831} \\ &= 44,5 \end{aligned}$$

Jadi nilai *PPM* berguna untuk mendapatkan nilai panjang x lebar dalam satuan millimeters seperti ukuran sebenarnya, dalam sistem ini menggunakan *PPM* sebesar 2.831 untuk mendapatkan data yang sesuai.

### 5.2.3.3 Implementasi Perancangan Klasifikasi *Naïve Bayes*

Pada implementasi selanjutnya akan dilakukan pengambilan data yang berasal dari pembacaan citra keripik kentang yang nantinya akan diklasifikasi menggunakan metode *Naïve Bayes*. Setelah data wmm dan hmm yang sudah dijelaskan pada subbab 0 didapatkan maka nilai siap untuk diklasifikasi. Pada Tabel 5.17 adalah potongan program untuk memasukkan nilai wmm dan hmm ke dalam array untuk diolah menggunakan *Naïve Bayes*.

**Tabel 5.17 Pengambilan data uji keripik kentang**

Baris	Kode Program
1	<code>datauji = []</code>
2	<code>datauji.append(wmm)</code>
3	<code>datauji.append(hmm)</code>

Implementasi program menggunakan metode *Naïve Bayes* bertujuan untuk mengklasifikasi jenis keripik kentang berdasarkan kelasnya dari data uji yang didapatkan pada pengambilan data sebelumnya.

Tahapan pertama dalam menggunakan program metode *Naïve Bayes* ialah menginisialisasi variabel untuk data latih seperti yang dituliskan pada Tabel 5.16. pertama adalah membuat variable prior, pada baris ke-1, menginisialisasi variable "prior" yang disimpan dalam array untuk menyimpan prior dari ketiga fitur dengan indeks pertama adalah prior kentang A, kedua AB dan ketiga adalah Super. Kemudian pada baris ke-4 variabel "mean\_std" yaitu menginisialisasi variable tersebut berupa array 3 dimensi, dimensi pertama adalah kelas jenis kentang A, AB dan Super dimensi kedua adalah terdapat 2 indeks, indeks ke-0 adalah mean dan indeks ke-1 adalah standard deviasi. Kemudian dimensi ketiga adalah nilai mean dan standard deviasi pada fitur tiap kelas. Lalu pada baris ke-23 adalah menginisialisasi variable "hasil\_text" yang berupa array untuk hasil klasifikasi kelas keripik kentang.

**Tabel 5.18 Kode program inisialisasi variabel untuk klasifikasi *Naive Bayes***

Baris	Kode Program
1	<code>#inisialisasi data latih naïve bayes</code>
2	<code>prior = [</code>
3	<code>0.333, # class "A" : 30x percobaan / 90 data</code>
4	<code>0.333, # class "AB" : 30x percobaan / 90 data</code>
5	<code>0.333 # class "Super" : 30x percobaan / 90</code>
6	<code>]</code>
7	<code>mean_std = [</code>
8	<code>[ # class "A"</code>
9	<code>[26.95160721, 30.778288], #mean KENTANG A</code>
10	<code>[5.159966, 5.142077] #standart deviasi KENTANG A</code>
11	<code>],</code>
12	<code>[ # class "AB"</code>
13	<code>[37.23066054, 40.11538914], #mean KENTANG AB</code>
14	<code>[6.50442, 6.391709] #standart deviasi KENTANG AB</code>



15	],
16	[ # class "SUPER"
17	[49.11103262, 57.2118215], #mean KENTANG SUPER
18	[16.29323, 13.90258] #standart deviasi KENTANG
19	SUPER
20	]
21	]
22	hasil_text = ["KENTANG A", "KENTANG AB", "KENTANG
23	SUPER"]

Setelah melakukan inisialisasi, membuat fungsi "Gaussian" yang dapat dilihat pada Tabel 5.19. Perhitungan nilai Gaussian didapat dari dalam variabel data latih yang telah diinisialisasi pada Implementasi program menggunakan metode *Naïve Bayes* bertujuan untuk mengklasifikasi jenis keripik kentang berdasarkan kelasnya dari data uji yang didapatkan pada pengambilan data sebelumnya.

Tahapan pertama dalam menggunakan program metode *Naive Bayes* ialah menginisialisasi variabel untuk data latih seperti yang dituliskan pada Tabel 5.16. pertama adalah membuat variable prior, pada baris ke-1, menginisialisasi variable "prior" yang disimpan dalam array untuk menyimpan prior dari ketiga fitur dengan indeks pertama adalah prior kentang A, kedua AB dan ketiga adalah Super. Kemudian pada baris ke-4 variabel "mean\_std" yaitu menginisialisasi variable tersebut berupa array 3 dimensi, dimensi pertama adalah kelas jenis kentang A, AB dan Super dimensi kedua adalah terdapat 2 indeks , indeks ke-0 adalah mean dan indeks ke-1 adalah standard deviasi. Kemudian domensi ketiga adalah nilai mean dan standard deviasi pada fitur tiap kelas. Lalu pada baris ke-23 adalah menginisialisasi variable "hasil\_text" yang berupa array untuk hasil klasifikasi kelas keripik kentang.

Tabel 5.18Tabel 5.19 Baris ke-1 adalah fungsi Gaussian dengan parameter data uji dan data latih tiap kelas jenis keripik kentang. Pada awalnya diinisialisasi dahulu variabel "hasil\_gaussian" untuk menyimpan hasil perhitungan Gaussian nantinya. Pada baris ke 3-10 menunjukkan perhitungan Gaussian secara berulang hingga seluruh fitur didapatkan hasil gaussiannya, yang kemudian hasilnya disimpan dalam sebuah array pada variabel "hasil\_gaussian".

**Tabel 5.19 Kode Program Fungsi Gaussian**

Baris	Kode Program
1	def gaussian(datauji, datalatih):
2	hasil_gaussian = []
3	for x in range(0,2):
4	a = 2 * 3.14 * (pow(datalatih[1][x], 2))
5	b = - ((pow((datauji[x] - datalatih[0][x]),
6	2)) / (2 * pow(datalatih[1][x],2)))
7	c = pow(2.718282, b)
8	d = 1 / math.sqrt(a)
9	
10	hasil_gaussian.append(c * d)
11	return hasil_gaussian"]



Kemudian dilanjutkan dengan implementasi kode program untuk fungsi “posterior”. Nilai posterior didapat dari perkalian nilai gaussian keseluruhan dengan prior yang dituliskan dalam diagram alir pada Gambar 5.7. Pada Tabel 5.20 ditunjukkan tabel fungsi “posterior”. Pada baris ke-1 fungsi probPosterior membutuhkan 2 parameter yaitu prior dan hasil\_gaussian. Pada baris kedua variabel “hasil” diinisialisasi sama dengan nilai prior. Pada baris ke-3 sampai dengan ke-5 menunjukkan perkalian secara berulang hingga semua fitur dikalikan, kemudian disimpan dalam variabel hasil.

**Tabel 5.20 Kode program fungsi posterior**

Baris	Kode Program
1	<code>def probPosterior(prior, hasil_gaussian):</code>
2	<code>    hasil = prior</code>
3	<code>    for x in range(0,2):</code>
4	<code>        hasil = hasil * hasil_gaussian[x] * 1000</code>
5	<code>        # dikali 1000 agar angka dibelakang kom tidak</code>
6	<code>hilang</code>
7	<code>    return hasil</code>

Setelah semua tahapan klasifikasi dilakukan maka dilakukan proses penyimpulan untuk mendapatkan hasil peluang tertinggi. Kode program fungsi kesimpulan dapat dilihat pada Tabel 5.21 Pada baris ke-1 fungsi kesimpulan mempunyai parameter hasil, kemudian pada baris ke-2 menentukan index yang mempunyai nilai tertinggi kemudian juga dimasukkan dalam variabel hasil\_text pada indeks berapa nilai terbesar apakah pada index kentang, AB atau Super. Pada baris ke-3 adalah mengirim index yang paling tinggi untuk memberi perintah mendorong aktuator servo ke arduino.

**Tabel 5.21 Kode program fungsi kesimpulan**

Baris	Kode Program
1	<code>def kesimpulan(hasil):</code>
2	<code>    max_index = hasil.index(max(hasil))</code>
3	<code>    bus.write_byte(0x24,max_index+1)</code>
4	<code>    return hasil_text[max_index]</code>

#### 5.2.3.4 Implementasi Kode Program Servo Pendorong Kentang

Setelah keripik kentang diklasifikasi, tahap selanjutnya yaitu pengklasifikasian yang dilakukan oleh aktuator berupa lengan pendorong yang digerakkan oleh servo. Pada Tabel 5.22 adalah program untuk memilah kentang sesuai kelasnya. Pada baris ke-1 sampai baris ke-3 adalah inisialisasi servo 1 sampai dengan 3, lalu pada baris ke-5 adalah menginisialisasi servo 1 sampai dengan 3 dengan type data bool false, baris ke 7-8 adalah deklarasi beberapa variabel bertipe unsigned long. Pada baris ke-12 adalah inisialisasi alamat untuk *interface* I2C dan ketika menerima data, baris ke-14 sampai dengan 16 adalah inisialisasi pin servo pada arduino yaitu pada baris ke 3, 5 dan 6, pada baris ke-18 sampai baris ke-20

mengeset servo dengan sudut 90 derajat. Pada perulangan pada baris 26 sampai 54 adalah menentukan kapan servo bergerak sebesar 180 derajat dan kembali ke posisi awal yaitu 90 derajat. Waktu servo bergerak diatur dalam fungsi "receiveData" yang dituliskan pada baris ke-58 sampai dengan baris ke-76.

**Tabel 5.22 Kode program servo**

Baris	Kode Program
1	Servo myservo1;
2	Servo myservo2;
3	Servo myservo3;
4	
5	bool servol = false, servo2 = false, servo3 = false;
6	
7	unsigned long WAIT_A_UP, WAIT_A_DOWN, WAIT_B_UP,
8	WAIT_B_DOWN, WAIT_C_UP, WAIT_C_DOWN;
9	
10	void setup() {
11	// put your setup code here, to run once:
12	Wire.begin(SLAVE_ADDR);
13	Wire.onReceive(receiveData);
14	
15	myservo1.attach(3); // servo pin 3,5,6
16	myservo2.attach(5);
17	myservo3.attach(6);
18	myservo1.write(90);
19	myservo2.write(90);
20	myservo3.write(90);
21	
22	Serial.begin(9600);
23	}
24	
25	void loop() {
26	if (servol == true){
27	if (millis() >= WAIT_A_DOWN){
28	Serial.println("Servo 1 Down");
29	myservo1.write(180);
30	}
31	if (millis() >= WAIT_A_UP){
32	myservo1.write(90);
33	servol = false;
34	Serial.println("Servo 1 Up");
35	}
36	if (servo2 == true){
37	if (millis() >= WAIT_B_DOWN){
38	Serial.println("Servo 2 Down");
39	myservo2.write(180);
40	}
41	if (millis() >= WAIT_B_UP){
42	myservo2.write(90);
43	servo2 = false;

```

44     Serial.println("Servo 2 Up");
45   }
46 }
47 if (servo3 == true){
48   if (millis() >= WAIT_C_DOWN){
49     Serial.println("Servo 3 Down");
50     myservo3.write(180);
51   }
52   if (millis() >= WAIT_C_UP){
53     myservo3.write(90);
54     servo3 = false;
55     Serial.println("Servo 3 Up");
56   }
57 }
58 void receiveData(int byteCount){
59   while(Wire.available()) {
60     number = Wire.read();
61
62     if (number == 1){
63       servo1 = true;
64       WAIT_A_DOWN = millis() + 1600;
65       WAIT_A_UP = WAIT_A_DOWN + 1000;
66     } else if(number == 2){
67       servo2 = true;
68       WAIT_B_DOWN = millis() + 3100;
69       WAIT_B_UP = WAIT_B_DOWN + 1000;
70     } else if (number == 3){
71       servo3 = true;
72       WAIT_C_DOWN = millis() + 4600;
73       WAIT_C_UP = WAIT_C_DOWN + 1000;
74     }
75     number = 0;
76   }
77 }

```

Dibawah ini adalah perhitungan manual pada saat mendorong servo yang mempunyai jarak dari kamera adalah servo 1 adalah 12,7 cm ,servo 2 adalah 24,7 cm dan servo 3 adalah 36,7. Mulanya 1 step pada stepper membutuhkan waktu  $2500 \times 2 = 5000$  microsecond yang berarti 5 ms, 1 revolusi stepper mempunyai 200 step maka untuk menghitung kecepatan belt stepper adalah

$$\text{Kecepatan Stepper} = \frac{\text{jumlah 1 revolusi stepper} \cdot \text{waktu 1 step}}{1000}$$

$$\text{Kecepatan Stepper} = \frac{200 \cdot 5}{1000} = 1 \text{ rpd (round per detik)}$$

Kemudian keliling lingkaran pada roller belt konveyor adalah :

$$\text{Kel roller} = 2\pi \cdot r = \pi \cdot d$$

$$\text{Kel roller} = 3,14 \times 2,54 = 7,97 \text{ cm}$$

Untuk menghitung jarak antara area pengambilan data setelah dikomputasi dengan servo adalah :

$$\text{Servo 1} = 12,7 \div 7,97 = 1,6 \text{ s atau } 1600 \text{ ms}$$

$$\text{Servo 2} = 24,7 \div 7,97 = 3,1 \text{ s atau } 3100 \text{ ms}$$

$$\text{Servo 3} = 36,7 \div 7,97 = 4,6 \text{ s atau } 4600 \text{ ms}$$

Jadi saat mendorong keripik kentang sesuai dengan kelasnya membutuhkan delay masing-masing sebanyak 1600, 3100 dan 4600 ms.



## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini berisi tentang pengujian dan analisis hasil pengujian dari system yang telah dirancang. Tujuan dilakukannya pengujian adalah mengetahui tingkat keberhasilan sistem bekerja sesuai dengan apa yang diinginkan.

### 6.1 Pengujian Pengambilan Nilai Objek Keripik Kentang Menggunakan *Background Subtraction*

Tujuan dilakukan pengujian ini adalah untuk mengetahui apakah sistem dapat membaca dan mengolah citra kentang menggunakan *Background Subtraction*. Sistem akan mengambil nilai yang digunakan untuk diklasifikasi yaitu nilai *boundingbox*  $w$  dan  $h$  dari biner yang sudah dikonversi ke ukuran sebenarnya menjadi  $wmm$  dan  $hmm$  atau panjang dan lebar *box* tersebut.

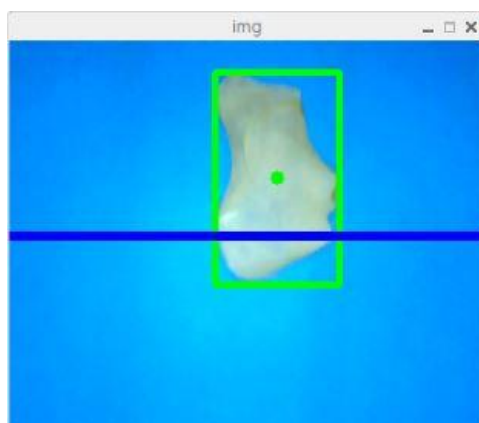
#### 6.1.1 Prosedur pengujian

Berikut prosedur yang dilakukan untuk menguji pengambilan data dari keripik kentang menggunakan *Background Subtraction*.

1. Menghubungkan *power supply* ke saluran listrik secara otomatis semua komponen akan menyala
2. *Upload* program ke Arduino Uno
3. Memasukkan program untuk memproses gambar pada Raspberry Pi
4. Jalankan program dari Arduino dan Raspberry
5. Memasukkan kentang ke konveyor secara vertical atau horizontal
6. Mengamati pembacaan tersebut apakah data dari kentang dapat terbaca oleh sistem, dan mencetak nilai  $wmm$  dan  $hmm$  dari keripik kentang tersebut.

#### 6.1.2 Hasil dan Analisis Pengujian

Hasil pengujian pengambilan gambar dari keripik kentang menggunakan *Background Subtraction* ditunjukkan pada Gambar 6.1 sampai dengan 6.3. Gambar dibawah menunjukkan pembacaan data kentang menggunakan *Background Subtraction*.



Gambar 6.1 Gambar frame yang diambil



Gambar 6.2 Gambar Biner frame Background Subtraction dan Opening



Gambar 6.3 Screenshoot data yang diambil dari bounding box keripik kentang

Berdasarkan pada Gambar 6.1, Gambar 6.2 dan Gambar 6.3 metode pengolahan gambar menggunakan *Background Subtraction* dapat mengambil data objek berupa keripik kentang dengan baik. *Belt* dari konveyor dibaca sebagai *background* dan kentang dibaca sebagai *foreground*.

## 6.2 Pengujian Pengambilan Gambar Keripik Kentang dari Sistem

Tujuan dilakukan pengujian ini adalah mengetahui tingkat akurasi sistem dalam membaca ukuran dari objek keripik kentang. Yaitu dengan cara membandingkan nilai yang dibaca oleh sistem dengan nilai dari alat ukur manual



berupa penggaris, dari situ akan diperoleh selisih antara pembacaan sensor yang kemudian dapat dihitung nilai *error* dari sistem pengambilan data objek gambar yang diterapkan.

### 6.2.1 Prosedur Pengujian

Berikut prosedur yang dilakukan untuk menguji pengambilan data dari gambar keripik kentang.

1. Menghubungkan power supply ke saluran listrik secara otomatis semua komponen akan menyala
2. *Upload* program ke Arduino Uno
3. Memasukkan program untuk memproses gambar pada Raspberry Pi
4. Jalankan program dari Arduino dan Raspberry
5. Mengukur beberapa keripik kentang dengan menggunakan penggaris
6. Mengamati hasil pembacaan gambar dari keripik kentang dari Raspberry Pi dan membandingkan nilai yang terbaca tersebut dengan nilai dari alat ukur penggaris untuk menentukan besarnya error pembacaan data gambar dari webcam di Raspberry.
7. Nilai panjang dan lebar (wmm dan hmm) masing-masing pengukuran kemudian diambil selisihnya untuk dilakukan persentase *error*.

Cara untuk mengukur persentase *error* yaitu dengan menggunakan **Persamaan (6.2)** berikut :

$$\text{Persentase error} = \frac{\text{Selisih nilai pembacaan}}{\text{Pembacaan alat ukur}} \times 100\% \quad (6.1)$$

Untuk menghitung nilai selisih pembacaan nilai dari pengolahan gambar dengan alat ukur yaitu dengan menggunakan **Persamaan (6.3)** berikut :

$$\text{Selisih nilai pembacaan} = |\text{pembacaan alat ukur} - \text{pembacaan webcam}| \quad (6.2)$$

### 6.2.2 Hasil dan Analisis Pengujian

Hasil dari pengujian pengambilan gambar dari keripik kentang menggunakan webcam yang diolah menggunakan *Background Subtraction* dalam Raspberry dituliskan pada Tabel 6.1. Pada tabel tersebut terlihat bahwa sistem membaca beberapa bentuk kentang yang berbeda. Hasil perbandingan antara nilai pembacaan citra dan dari alat ukur pada objek keripik kentang yang sama menunjukkan perbedaan nilai.

**Tabel 6.1 Hasil pengujian pembacaan kentang dari sistem**

Pengujian ke	Pembacaan Gambar (milimeter)			Alat Ukur (milimeter)			Selisih Error (milimeter)	Error
	Hmm	Wmm	X	Hmm	Wmm	x	x	
1	60,5	38,14	2307,47	61	38	2318	10,53	0.45%
2	58,5	41,68	2438,28	59	42	2478	39,2	1,58%
3	57,22	43,09	2465.60	59	43	2537	71,4	2,81%
4	44,86	28,96	1299.14	46	30	1380	80,86	5.85%
5	42,38	32,49	1376.92	43	34	1462	85.08	5,81%
6	50,2	46,73	2345.84	50	46	2300	45.84	1,99%
7	27,19	23,17	629.99	28	24	672	42.01	6.25%
8	32,95	31,37	1033.64	34	33	1122	88,36	7,87%
9	33,91	20,48	694.47	34	21	714	19,53	2,74%
10	32,49	21,90	711.531	33	22	726	14,47	1,99%
Rata-rata								3,73 %

Berdasarkan **Persamaan (6.1)** untuk menentukan persentase *error* pada tiap pengujian, diperoleh rata-rata *error* sebesar 3,73 %. Contoh perhitungan persentase *error* pada pengujian ke-1 adalah sebagai berikut:

Nilai perkalian *Wmm* dan *Hmm* dari pengolahan gambar = 2307,47

Nilai perkalian *Wmm* dan *Hmm* dari alat ukur = 2318

$$\begin{aligned} \text{Selisih nilai pembacaan} &= |\text{pembacaan alat ukur} - \text{pembacaan citra}| \\ &= 2318 - 2307,47 = \mathbf{10,53} \end{aligned}$$

$$\begin{aligned} \text{Persentase error} &= \frac{\text{Selisih nilai pembacaan}}{\text{Pembacaan alat ukur}} \times 100\% \\ &= \frac{\mathbf{10,53}}{2318} \times 100\% \\ &= 0.45\% \end{aligned}$$

Perhitungan menghitung rata-rata *error* keseluruhan pengujian adalah sebagai berikut :

$$\begin{aligned} \text{Rata - rata Persentase error} &= \frac{\text{Jumlah persentase error}}{\text{Jumlah pengujian}} \\ &= \frac{0.45\% + 1,58\% + 2,81\% + 5.85\% + 5,81\% + 1,99\% + 6.25\% + 7,87\% + 2,74\% + 1,99\%}{10} \\ &= \mathbf{3,73\%} \end{aligned}$$

Rata-rata *error* yang diperoleh pengolahan gambar sangat kecil, sehingga dapat disimpulkan bahwa akurasi pembacaan citra adalah baik. Error terjadi karena kentang pada posisi tidak sempurna *vertical* karena pada saat dimasukkan, biasanya keripik berubah posisi sedikit yang berdampak pada pengambilan data pada citra kentang dan pada alat ukur tidak bisa menghitung koma juga berdampak pada persentase *error*.

### 6.3 Pengujian Akurasi Hasil Klasifikasi *Naïve Bayes*

Implementasi *Background Subtraction* Untuk Klasifikasi Keripik Kentang Berbasis Raspberry Pi menggunakan Metode *Naïve Bayes* mempunyai tujuan utama yaitu mengklasifikasi jenis keripik kentang sesuai dengan kelasnya melalui ukurannya, sehingga perlu diketahui tingkat akurasi sistem dalam melakukan pengklasifikasian.

#### 6.3.1 Tujuan Pegujian

Tujuan dilakukan pengujian ini adalah untuk menentukan tingkat akurasi menggunakan metode *Naïve Bayes* pada sistem yang telah dirancang.

#### 6.3.2 Prosedur Pegujian

Prosedur pengujian akurasi menggunakan metode *Naïve Bayes* dilakukan dengan adanya masukan 2 fitur pembacaan objek keripik kentang menggunakan perbandingan antara data latih dan data uji dengan rasio sebesar 3:1. Penentuan proporsi data latih lebih banyak agar akurasi yang didapat semakin tinggi. Dimana dari data latih diambil 30 sampel sesuai kelasnya yaitu A, AB dan Super jadi totalnya adalah 90, kemudian mengambil lagi dari masing-masing kelas 10 keripik kentang untuk data uji yang totalnya adalah 30 data. Cara yang digunakan dalam menentukan nilai akurasi dari sistem yakni dengan membandingkan hasil pengklasifikasian dari sistem dengan hasil keripik yang sudah disortir yang didapatkan dari pabrik keripik kentang Istana Kota Batu pada proses pengumpulan data. Untuk menghitung akurasi digunakan **Persamaan (6.3)**.

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ data} \times 100\% \quad (6.3)$$

#### 6.3.3 Hasil dan Analisis Pengujian

**Tabel 6.2 Data uji dan hasil pengujian *Naïve Bayes***

No	Wmm (milimeter)	Hmm (milimeter)	Kelas	Hasil Sistem	Kesesuaian
1	24,726	26,845	A	A	Sesuai
2	21,193	27,198	A	A	Sesuai
3	19,427	27,198	A	A	Sesuai

4	19,074	26,845	A	A	Sesuai
5	26,139	27,196	A	A	Sesuai
6	21,193	33,203	A	A	Sesuai
7	23,313	30,024	A	A	Sesuai
8	26,845	30,024	A	A	Sesuai
9	31,790	22,606	A	A	Sesuai
10	19,074	33,910	A	A	Sesuai
11	37,442	44,507	AB	AB	Sesuai
12	39,561	43,094	AB	AB	Sesuai
13	37,442	47,333	AB	AB	Sesuai
14	36,736	46,626	AB	AB	Sesuai
15	39,561	43,094	AB	AB	Sesuai
16	51,571	69,233	AB	SUPER	Tidak Sesuai
17	48,636	37,467	AB	AB	Sesuai
18	42,984	36,999	AB	AB	Sesuai
19	37,552	41,193	AB	AB	Sesuai
20	40,865	38,527	AB	AB	Sesuai
21	53,691	69,233	SUPER	SUPER	Sesuai
22	41,681	58,283	SUPER	SUPER	Sesuai
23	37,442	63,935	SUPER	SUPER	Sesuai
24	45,213	59,696	SUPER	SUPER	Sesuai
25	45,920	54,397	SUPER	SUPER	Sesuai
26	41,681	58,989	SUPER	SUPER	Sesuai
27	40,974	62,522	SUPER	SUPER	Sesuai
28	38,855	43,691	SUPER	AB	Tidak Sesuai
29	47,333	76,646	SUPER	SUPER	Sesuai
30	69,939	68,880	SUPER	SUPER	Sesuai

Hasil pengujian yang ditunjukkan pada Hasil dari pengujian pengambilan gambar dari keripik kentang menggunakan webcam yang diolah menggunakan *Background Subtraction* didalam Raspberry dituliskan pada Tabel 6.1. Pada tabel tersebut terlihat bahwa sistem membaca beberapa bentuk kentang yang berbeda.

Hasil perbandingan antara nilai pembacaan citra dan dari alat ukur pada objek keripik kentang yang sama menunjukkan perbedaan nilai.

2 dapat dilihat bahwa dari jumlah 30 data yang diambil, hanya sedikit keripik kentang yang tidak sesuai dengan kelasnya yaitu sebanyak 2. Sehingga akurasi yang diperoleh Sistem ini dengan metode *Naïve Bayes* ini adalah sebesar 93,33%. Proses perhitungan akurasi dapat dilihat sebagai berikut menggunakan persamaan **Persamaan (6.3)** :

$$\begin{aligned} \text{Akurasi} &= \frac{\text{Total data} - \text{Data tidak sesuai}}{\text{Total data}} \times 100\% \\ &= \frac{30 - 2}{30} \times 100\% \\ &= 93,33\% \end{aligned}$$

#### 6.4 Pengujian Servo Pendorong Keripik Kentang

Servo merupakan komponen atau actuator untuk menggerakkan pendorong untuk mengklasifikasikan keripik kentang. Servo bergerak berdasarkan output yang dikirimkan dari Raspberry Pi ke Arduino melalui komunikasi I2C. Oleh karena itu, pengujian ini dilakukan dengan melihat kesesuaian antara data yang dikirim Raspberry Pi dengan servo yang diproses Arduino Uno.

##### 6.4.1 Tujuan Pegujian

Pegujian Servo pendorong keripik kentang ini bertujuan untuk mengetahui apakah hasil dari klasifikasi dari sistem sesuai dengan servo aktuator pendorong kentang yang telah dibuat dan diimplementasikan.

##### 6.4.2 Prosedur Pengujian

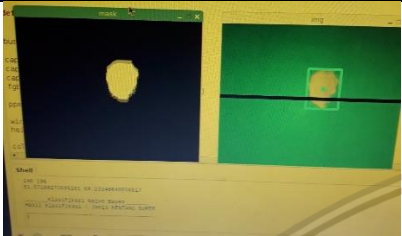

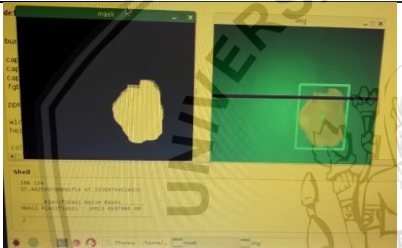

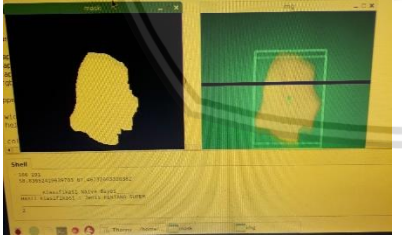

Untuk melakukan pengujian Servo pendorong keripik kentang dilakukan prosedur seperti berikut :

1. Menghubungkan power supply ke saluran listrik secara otomatis semua komponen akan menyala
2. *Upload* program ke Arduino Uno
3. Memasukkan program untuk memproses gambar pada Raspberry Pi
4. Jalankan program dari Arduino dan Raspberry
5. Masukkan keripik kentang ke konveyor secara vertical atau horizontal
6. Mengamati kentang berada pada kelas apa pada python, kemudian melihat apakah servo mendorong kentang sesuai dengan kelas yang sudah diklasifikasi sistem.



### 6.4.3 Hasil dan Analisis Pengujian

**Tabel 6.3 Hasil pengujian servo pendorong keripik kentang**

No	Hasil Klasifikasi	Servo Pendorong	Keterangan	Hasil
1			Sistem mengklasifikasi kentang A, Servo 1 mendorong kentang	Sesuai
2			Sistem mengklasifikasi kentang AB, Servo 2 mendorong kentang	Sesuai
3			Sistem mengklasifikasi kentang Super, Servo 3 mendorong kentang	Sesuai

Berdasarkan hasil pengujian servo pendorong keripik kentang yang ditunjukkan pada Tabel 6.3 terlihat bahwa servo dapat bergerak sesuai apa yang dikirimkan oleh Raspberry ke arduino.



## 6.5 Pengujian Waktu Komputasi Klasifikasi Pemrosesan Citra

### 6.5.1 Tujuan Pegujian

Tujuan dilakukan pengujian ini adalah untuk mengetahui waktu yang dibutuhkan untuk proses klasifikasi jenis keripik kentang menggunakan metode Naïve Bayes.

### 6.5.2 Prosedur Pegujian

Dalam melakukan pengujian mengenai waktu komputasi pemrosesan citra dilakukan dengan cara menghitung waktu komputasi ketika program mengambil data dari keripik kentang dalam satu siklus sebanyak 30 kali pengujian. Waktu komputasi diambil dari waktu komputasi akhir dikurangi komputasi awal dimulai dari data masuk sampai selesai diklasifikasi. Dalam implementasi program untuk komputasi pada sistem ini menggunakan fungsi “time” pada python yang berfungsi untuk mendapatkan waktu dalam satuan 1 *millisecond* (ms). Fungsi millis dimulai dari data wmm dan hmm masuk hingga selesai klasifikasi *Naïve Bayes* Implementasi millis() pada Python ditunjukkan Tabel 6.4.

**Tabel 6.4 Kode program fungsi “millis()”**

Baris	Kode Program
1	def millis():
2	return int (round(time.time()*1000))
3	.
4	.
5	.
6	Start_time = millis()
7	.
8	Print (millis() - start_time)

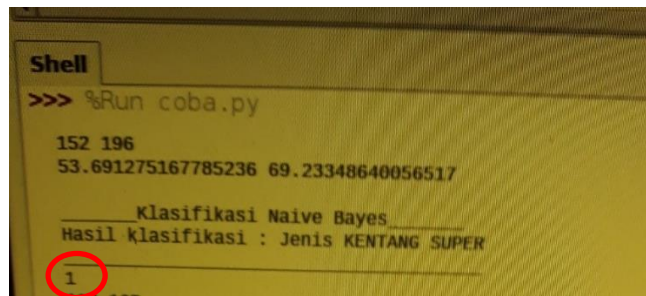
### 6.5.3 Hasil dan Analisis

**Tabel 6. 5 Pengujian Waktu Komputasi**

No	Pengujian ke-	Waktu Komputasi (ms)
1	1	2
2	2	1
3	3	1
4	4	1
5	5	2
6	6	3
7	7	1
8	8	1

9	9	2
10	10	2
11	11	1
12	12	3
13	13	3
14	14	2
15	15	1
16	16	3
17	17	2
18	18	1
19	19	1
20	20	3
21	21	1
22	22	1
23	23	2
24	24	1
25	25	2
26	26	1
27	27	1
28	28	3
29	29	2
30	30	2
<b>Rata-rata</b>		<b>1,7</b>

Berdasarkan hasil pengujian yang dilakukan sebanyak 30 kali, waktu komputasi sistem untuk melakukan pengambilan keputusan jenis keripik kentang adalah dengan rata-rata 1,7 ms. Screenshoot salah satu pengujian waktu komputasi pada python ditunjukkan pada Gambar 6.4.



Gambar 6.4 Hasil pengujian waktu komputasi



## BAB 7 PENUTUP

Pada bab ini berisi tentang penarikan kesimpulan dari pengujian yang dilakukan. Kemudian memnulkan beberapa saran untuk pengembangan penelitian selanjutnya.

### 7.1 Kesimpulan

Berdasarkan rumusan masalah yang dituliskan pada awal pengerjaan dan juga hasil analisis dan pengujian yang dilakukan maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Pengambilan Nilai Objek Keripik Kentang Menggunakan *Background Subtraction* dan Menghilangkan Noise dengan *Opening* dapat terbaca dengan baik sesuai dengan apa yang diinginkan. Sehingga dapat simpulkan *Background Subtraction* dapat membaca nilai kentang yang berjalan diatas belt konveyor melewati webcam.
2. Pembacaan ukuran keripik kentang dari sistem mendapatkan *error* yang kecil yaitu sebesar 3,73%, sehingga dapat dinyatakan sistem dapat membaca nilai dari keripik kentang dengan baik karena dapat membaca ukuran kentang sesuai dengan aslinya.
3. Pada penelitian ini data dari keripik kentang yang didapat diklasifikasi menggunakan metode *Naïve Bayes*. Keripik kentang dapat dibedakan sesuai dengan kelasnya yaitu kelas A, AB dan Super. Akurasi yang diperoleh menggunakan metode *Naïve Bayes* dalam klasifikasi keripik dengan data latih 90 dan data uji 30 adalah senilai 93,33 sehingga dapat dikatakan metode *Naïve Bayes* dapat mengklasifikasi keripik kentang dengan baik sesuai dengan kelasnya.
4. Pada sistem ini aktuator pemilah atau pendorong kentang adalah berupa pendorong yang dijalankan oleh servo berdasarkan apa yang dikirimkan oleh Raspberry Pi ke arduino. Dimana dalam sistem ini servo dapat menggerakkan pendorong yang mengklasifikasi kentang sesuai dengan kelasnya. Sehingga dapat dikatakan servo bekerja sesuai dengan apa yang diharapkan memilah kentang sesuai dengan kelasnya.
5. Performansi Implementasi *Background Subtraction* untuk Klasifikasi Keripik Kentang Berbasis Raspberry Pi Menggunakan Metode *Naïve Bayes* mempunyai waktu pemrosesan rata-rata sebesar 1,7 ms dari 30 kali dilakukan pengujian.

### 7.2 Saran

Setelah dilakukan penelitian, ada beberapa saran yang dapat digunakan untuk pengembangan penelitian kedepan sebagai berikut:

1. Menggunakan metode pengolahan citra lain untuk membandingkan pengolahan citra manakah yang mempunyai peforma yang lebih baik.
2. Menggunakan metode atau algoritma klasifikasi lain untuk membandingkan metode atau algoritma manakah yang mempunyai keakuratan yang lebih tinggi.
3. Menggunakan kamera webcam yang mempunyai spesifikasi lebih baik untuk lebih mendapatkan gambar yang baik.
4. Menerapkan sistem serupa untuk memilah objek lain.



## DAFTAR PUSTAKA

- Ardhianto, 2013. Implementasi Metode Image Substracting dan Metode regionprops untuk Mendeteksi Jumlah Objek Berwarna RGB pada File Video. *Jurnal Teknologi Informasi DINAMIK*, Volume 18, p. 2.
- Arduino. (2016). Arduino Uno Atmega328P.[Online] <<https://www.arduino.cc/en/Main/ArduinoBoardUno>>. [Diakses 20 April 2018]
- Arga Anugrahandy, B. D. A. B. S., 2013. Perancangan Alat Sortasi Otomatis Buah Apel Manalagi (*Malus sylvestris* Mill) Menggunakan Mikrokontroler AVR ATmega 16. *Jurnal Keteknikan Pertanian Tropis dan Biosistem*, Volume 1, pp. 1-9.
- Astuti, E. H. (2016). *Sistem Pendukung Keputusan Deteksi Dini Penyakit Stroke menggunakan Metode Naive Bayes*. SKRIPSI, Universitas Brawijaya, Teknik Informatika, Malang.
- Barber, D., 2010. *Bayesian Reasoning and Machine Learning*. London: s.n.
- Devi, 2018. *Klasifikasi Ukuran Irisan kentang*, Batu: s.n.
- Dias Prihatmoko, A. K. Z., 2015. SISTEM PENDETEKSI GERAK BERBASIS WEB MENGGUNAKAN. *DISPROTEK*, Volume 6, p. 20.
- ElectronicWings. Raspberry Pi GPIO Access  
.[Online] <<http://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access>>. [Diakses 5 Mei 2018]
- Izzati Saila Hafsah, P. N. A., 2015. Deteksi Otomatis Penyakit Kulit Menggunakan Algoritma Naive Bayes.
- Maulana, R., 2018. Implementasi Sistem Deteksi Slot Parkir Mobil Menggunakan Metode Morfologi dan Background Subtraction. *Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 1954-1959 .
- Pololu. A4988 Stepper Motor Driver Carrier.[Online] <<https://www.pololu.com/product/1182>>. [Diakses 20 April 2018]
- PotentialLabs. Tower Pro MG995 Metal Gear Servo.[Online] <<https://potentiallabs.com/cart/towerpro-mg995-servo>>. [Diakses 20 April 2018]
- Samir, M. I., n.d. Penerapan Algoritma Background Subtraction Untuk Tracking dan Klasifikasi Kendaraan.
- Sapermana, R., 2015. *Pengertian Citra dan Pengolahan Citra*. [Online] Available at: <http://www.romlisapermana.com/2015/07/pengertian-citra-dan-pengolahan-citra.html> [Diakses 18 April 2018].



Thiang., I. L., 2014. Otomasi Pemisah Buah Tomat Berdasarkan Ukuran Menggunakan Webcam Sebagai Sensor.

Zona Elektro.(2015). Teori Motor Stepper : Jenis dan Pronsip Motor Stepper.[Online] <<http://zoniaelektro.net/motor-stepper/>>. [Diakses 20 April 2018]

